

NEO Ninst



Авторы: Ахметов Канат, Новосёлов Андрей, Торопов Александр

Оглавление

Введение в мир NEONinst	3
Как пользоваться этой книгой?	3
Требования к окружению.....	3
Что используется в NeonInst?	4
JavaScript	4
CSS (Cascading Style Sheets).....	5
HTML (HyperText Markup Language)	5
Docker.....	5
FastAPI	5
PostgreSQL.....	5
SQLAlchemy	6
Целевая аудитория	6
Путешествие по функционалу	7
Основные понятия: все, что ты должен знать.....	8
Двигайся в ритме новостей и общения!	82

Введение в мир NEONinst

Добро пожаловать в NeonInst, где магия общения встречается с волнующим миром новостей! Мы не просто мессенджер, мы – твой личный гид в мире актуальной информации и удивительного общения. Готов погрузиться в это приключение?

Как пользоваться этой книгой?

Каждая глава книги построена таким образом, чтобы вы могли постепенно наращивать свои знания и навыки. В каждой главе вы найдете подробные инструкции, примеры кода и пояснения к ним.

Рекомендуется читать книгу последовательно, не пропуская главы, так как каждая следующая глава опирается на материал предыдущих. Однако, если вы уже обладаете определенными знаниями в некоторых областях, вы можете переходить к интересующим вас разделам.

Требования к окружению

Для работы над проектом вам потребуется установленный Python и текстовый редактор или IDE, такие как Visual Studio Code или PyCharm. Также понадобится браузер для тестирования фронтенда и установленный менеджер пакетов `pip` для установки необходимых библиотек.

Что используется в NeonInst?

В этом приложении было задействовано большое количество различных языков программирования, базы данных и т.д(тут ктонибудь допишите)!!!

Наше приложение делится на 2 части. Frontend и Backend

Python

Python — это высокоуровневый язык программирования общего назначения, созданный Гвидо ван Россумом и впервые выпущенный в 1991 году. Он известен своей простотой и читаемостью, что делает его доступным для начинающих и мощным для опытных программистов. Основные характеристики Python:

1. **Простота и читабельность:** Код на Python часто похож на псевдокод, благодаря минималистичному синтаксису и использованию отступов для структурирования блоков кода.
2. **Интерпретируемый язык:** Python выполняется интерпретатором, что облегчает отладку и тестирование кода.
3. **Многообразие парадигм:** Поддерживает процедурное, объектно-ориентированное и функциональное программирование.
4. **Большая стандартная библиотека:** Включает множество готовых к использованию модулей и пакетов для решения различных задач — от веб-разработки до научных вычислений.
5. **Платформенная независимость:** Python работает на различных операционных системах, включая Windows, macOS и Linux.

JavaScript

JavaScript — это высокоуровневый, динамический язык программирования, который в основном используется для создания интерактивных элементов на веб-страницах. Он работает на стороне клиента в веб-браузерах и позволяет разработчикам добавлять такие элементы, как анимации, формы обратной связи и другие интерактивные компоненты. JavaScript также может использоваться на стороне сервера благодаря таким платформам, как Node.js.

CSS (Cascading Style Sheets)

CSS — это язык стилей, используемый для описания внешнего вида и форматирования веб-страниц, написанных на HTML. CSS позволяет разработчикам управлять расположением, цветами, шрифтами и другими аспектами визуального представления элементов на веб-странице. Он поддерживает каскадность, что означает, что стили могут быть применены в зависимости от их приоритетов и источников.

HTML (HyperText Markup Language)

HTML — это язык разметки, который используется для создания и структурирования контента на веб-страницах. Он определяет структуру веб-страницы с помощью тегов, таких как `<h1>`, `<p>`, `<div>`, `<a>` и т.д. HTML обеспечивает основу, на которой построены все веб-документы, а также взаимодействует с CSS и JavaScript для создания полнофункциональных веб-приложений.

Docker

Docker — это платформа для контейнеризации, которая позволяет разработчикам упаковывать приложения и их зависимости в изолированные контейнеры. Эти контейнеры могут быть запущены на любой системе, поддерживающей Docker, обеспечивая совместимость и упрощая развертывание приложений. Docker облегчает создание, тестирование и развертывание приложений в различных средах, не беспокоясь о различиях в конфигурации.

FastAPI

FastAPI — это современный, быстрый (высокопроизводительный) веб-фреймворк для создания API на Python, основанный на стандарте OpenAPI и JSON Schema. Он разработан для обеспечения высокой скорости разработки, а также для обеспечения отличной производительности. FastAPI поддерживает асинхронные вызовы и имеет встроенную поддержку документации API.

PostgreSQL

PostgreSQL — это мощная, открытая реляционная система управления базами данных (СУБД). Она известна своей надежностью, полнотой функционала и расширяемостью. PostgreSQL поддерживает сложные запросы, транзакции, индексы и другие функции, которые делают её подходящей для различных приложений — от небольших сайтов до больших корпоративных систем.

SQLAlchemy

SQLAlchemy — это библиотека для Python, обеспечивающая полную функциональность работы с базами данных. Она предоставляет ORM (Object-Relational Mapping), который позволяет разработчикам взаимодействовать с базой данных с использованием объектов Python вместо прямых SQL-запросов. SQLAlchemy также поддерживает выражения SQLAlchemy Core для более низкоуровневого управления SQL-запросами.

Целевая аудитория

Мессенджер NeonInst имеет функционал, который удовлетворяет потребностям множеству групп пользователей, а именно:

- студенты и образовательные сообщества: для обмена информацией, обсуждения учебных материалов и получения новостей об актуальных событиях в образовании;
- профессиональные группы и команды: для коммуникации внутри организаций, обсуждения проектов и оперативного информирования о новостях в отрасли;
- любители новостей и общественно-политических дискуссий: для получения свежих новостей, обмена мнениями и участия в дискуссиях на различные темы;
- семейные группы и друзья: для обмена личными сообщениями, планирования мероприятий и получения обновлений от близких.

Путешествие по функционалу

- Регистрация и аутентификация

Для начала своего путешествия в NEONinst просто создай свой аккаунт и окунись в удивительный мир общения и новостей. Аутентификация здесь так же легка, как покачивание пальцами!

- Профиль

Твой профиль — это твоя визитная карточка в NEONinst. Добавь фото, расскажи немного о себе и готово! Подготовься к тому, чтобы все вокруг узнали о твоих интересах и талантах.

- Чаты

Пришло время веселиться в чатах! Общайся с друзьями, семьей или коллегами - в любое время и в любом месте. Здесь каждое обсуждение превращается в невероятное приключение!

- Новостная лента

И, конечно же, наша новостная лента — это твой персональный информационный портал. Будь в курсе всех событий, которые тебя интересуют, и делай свои выводы.

Основные понятия: все, что ты должен знать

Хочешь понять, как всё работает? Вот небольшой справочник, который поможет тебе сориентироваться в мире NEONinst:

- регистрация и аутентификация пользователей: здесь каждый новый пользователь — это как герой, который только начинает свое путешествие;
- профиль пользователя: твой паспорт в мир NEONinst. Здесь ты можешь показать всю свою индивидуальность и неповторимость;
- чаты: Место, где слова становятся магией общения. Здесь можно узнать много интересного и поделиться своими мыслями;
- новостная лента: твой информационный космос, где ты всегда будешь в курсе всех горячих событий;
- фронтенд (англ. front end, frontend) — это презентационная часть web-приложений, информационной или программной системы, её пользовательский интерфейс и связанные с ним компоненты.

Теперь, когда ты знаешь все основные понятия, давай начнем это удивительное путешествие вместе!

Авторизация.

HTML:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Авторизация</title>

  <link rel="stylesheet" href="{{ url_for('static', path='/style_auth.css')
}}">

</head>

<body>

  <div class="container">

    <form action="/login" method="post">

      <h1>Авторизация</h1>

      <div class="input-box">

        <input type="text" placeholder="Логин" name="username" required/>

      </div>

      <div class="input-box">

        <input type="text" placeholder="Пароль" name="password"
required/>

      </div>

      <button type="submit" class="btn">Войти</button>
```

```

        <div class="register-link">
            <p>Нет аккаунта? <a href="/registration">Зарегистрируйся
здесь!</a></p>
        </div>
    </form>
</div>

</body>
</html>

```

CSS:

```

@import
url('https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300..800;1,300..800&display=swap');

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: "Poppins";
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: url(../images/bck1.jpg) no-repeat;
    background-size: cover;
}

.container {
    /*background: green;*/

```

```
width: 420px;

border: 2px solid (255, 255, 255, 0.1);

box-shadow: 0px 0px 10px (0, 0, 0, 0.1);

color: #fff;

border-radius: 10px;

padding: 30px 40px;

backdrop-filter: blur(18px);

}
```

```
.container h1 {

    font-size: 36px;

    text-align: center;

}
```

```
.container .input-box{

    position: relative;

    width: 100%;

    height: 50px;

    margin: 30px 0px

}
```

```
.input-box input {

    width: 100%;

    height: 100%;

    background: transparent;

    border: none;

    outline: none;

    border: 2px solid (255, 255, 255, 0.2);

    border-radius: 5px;

    font-size: 16px;

    color: #fff;

    padding: 20px 45px 20px 20px;
```

```
}

.input-box input::placeholder {
    color: #fff;
}

.container .remember-forgot {
    display: flex;
    justify-content: space-between;
    font-size: 14.5px;
    margin: -15px 0px 15px;
}

.remember-forgot label input {
    color: #fff;
    margin-right: 3px;
}

.remember-forgot a{
    color: #fff;
    text-decoration: none;
}

.remember-forgot a:hover {
    text-decoration: underline;
}

.container .btn {
    width: 100%;
    height: 45px;
    background: #fff;
    border: none;
}
```



```

outline: none;

border-radius: 5px;

box-decoration-break: 0px 0px 10px rgba(0, 0, 0, 0.1);

cursor: pointer;

font-size: 16px;

font-weight: 400;

color: #333;
}

.container .register-link {

    font-size: 14.5px;

    text-align: center;

    margin: 20px 0px 20px;
}

.register-link p a {

    color: #fff;

    text-decoration: none;
}

.register-link p a:hover {

    text-decoration: underline;
}

```

PYTHON

```

@app.get("/auth", response_class=HTMLResponse)

def read_item(request: Request, usr_id: Union[str, None] =
Cookie(default=None)):

    global USER

    if usr_id is None:

        template = "auth.html"

```

```

        context = {"request": request}

        return templates.TemplateResponse(template, context)

    return RedirectResponse("/profile")

```

```

@app.post("/login")

def read_item(username: str = Form(...), password: str = Form(...),
db: Session = Depends(database.get_db), usr_id: Union[str, None] =
Cookie(default=None)):

    if usr_id is None:

        user = db.query(models.User).filter(models.User.username ==
username).first()

        if auth.verify_password(password, user.hashed_password):

            if user:

                usr_id = user.id

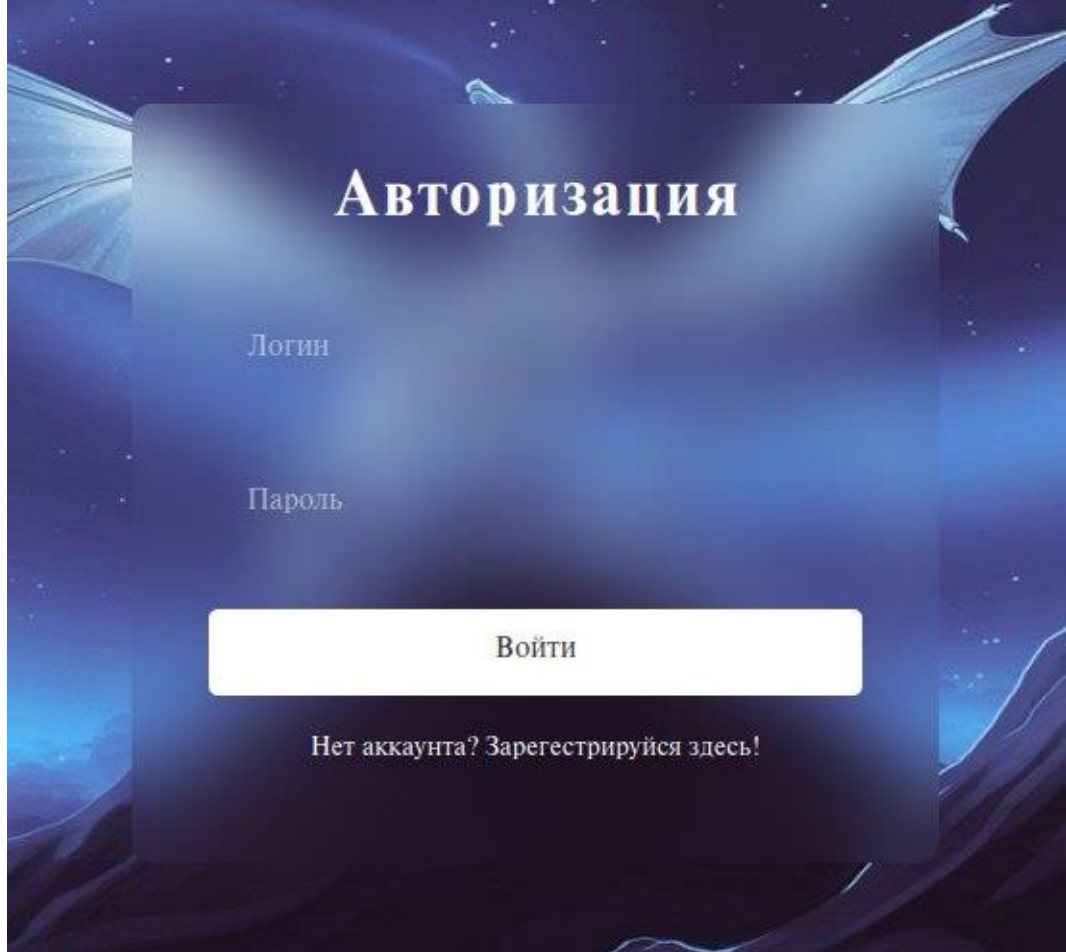
                resp = RedirectResponse("/profile", status_code=302)
                resp.set_cookie(key="usr_id", value=usr_id)

                return resp

    else:

        return RedirectResponse("/")

```



Регистрация:

HTML:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Регистрация</title>

    <link rel="stylesheet" href="{{ url_for('static',
path='/style_auth.css') }}">

</head>

<body>

    <div class="container">

        <form action="/register" method="post">

            <h1>Регистрация</h1>
```

```

        <div class="input-box">
            <input type="text" placeholder="Логин" name="username"
required/>
        </div>

        <div class="input-box">
            <input type="text" placeholder="Е-mail" name="email"
required/>
        </div>

        <div class="input-box">
            <input type="text" placeholder="Пароль" name="password"
required/>
        </div>

        <button type="submit" class="btn">Войти</button>

        <div class="register-link">
            <p>Есть аккаунт? <a href="/auth">Войдите здесь!</a></p>
        </div>
    </form>
</div>

</body>
</html>

```

В регистрации используется точно такой же файл стилей.

PYTHON

```

# Страница регистрации

@app.get("/registration", response_class=HTMLResponse)

```



```

def read_item(request: Request, usr_id: Union[str, None] =
Cookie(default=None)):

    if usr_id is None:

        template = "registration.html"

        context = {"request": request}

        return templates.TemplateResponse(template, context)

    return RedirectResponse("/profile")

```

```

# Регистрация пользователя

@app.post("/register") # response_model=schemas.User

def read_item(username: str = Form(...), email: str = Form(...), password:
str = Form(...), db: Session = Depends(database.get_db), usr_id:
Union[str, None] = Cookie(default=None)):

    if usr_id is None:

        db_user = db.query(models.User).filter(models.User.username ==
username).first()

        if db_user:

            raise HTTPException(status_code=400, detail="Username already
registered")

        hashed_password = auth.get_password_hash(password)

        db_user = models.User(username=username, email=email,
hashed_password=hashed_password)

        db.add(db_user)

        db.commit()

        db.refresh(db_user)

    # Получаем модель нашего пользователя

    user = db.query(models.User).filter(models.User.username ==
username).first()

    if user:

        usr_id = user.id

```

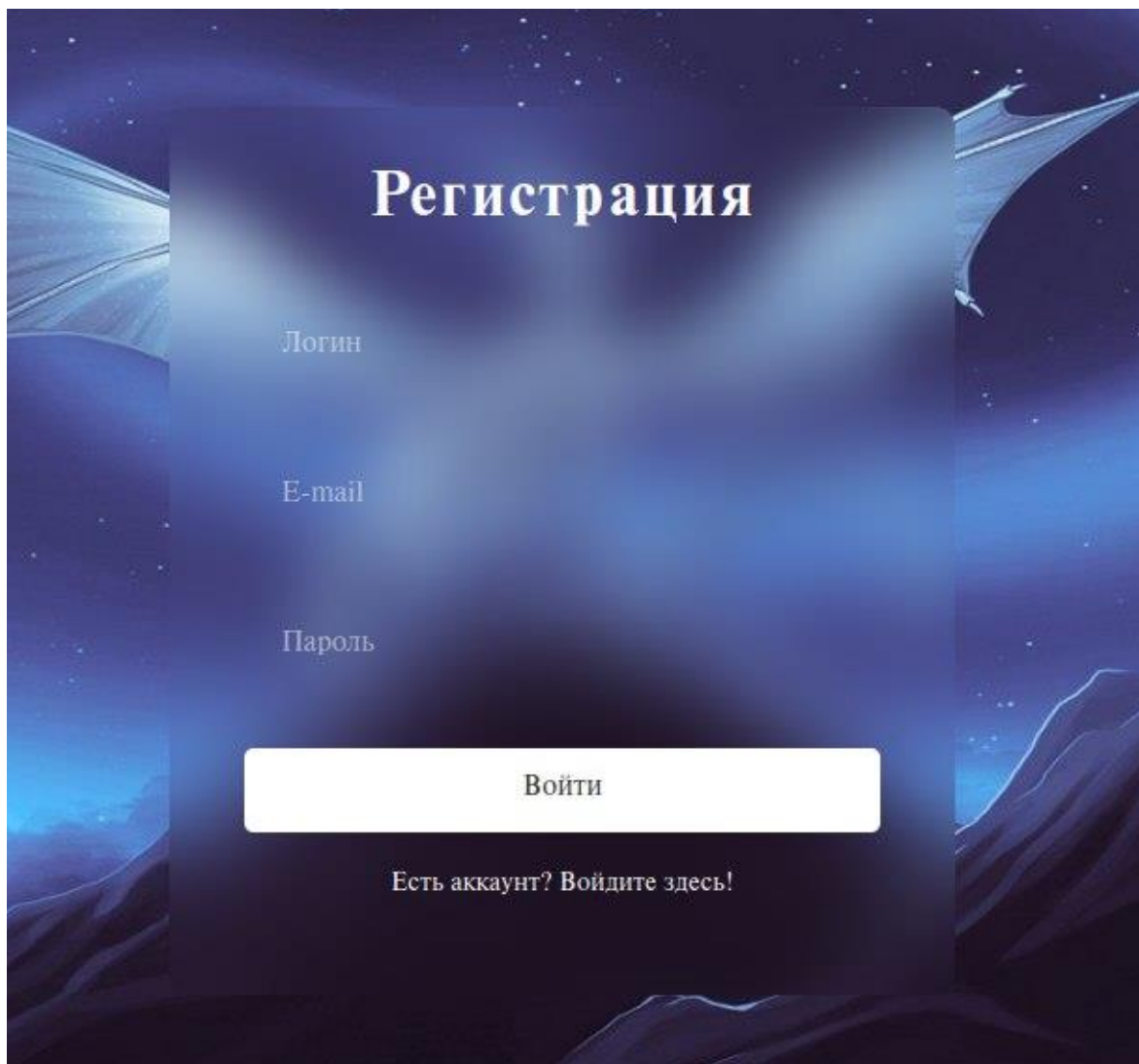
```
resp = HttpResponseRedirect("/profile", status_code=302)

resp.set_cookie(key="usr_id", value=usr_id)

return resp

else:

return HttpResponseRedirect("/")
```



The image shows a registration form titled "Регистрация" (Registration) set against a dark blue, space-themed background with star trails and mountain silhouettes. The form is a dark gray rectangle containing three input fields labeled "Логин" (Login), "E-mail", and "Пароль" (Password). Below these fields is a white button with the text "Войти" (Login). At the bottom of the form, there is a link that says "Есть аккаунт? Войдите здесь!" (Have an account? Log in here!).

В результате, данная страница создает форму для регистрации нового пользователя с полями для ввода логина, email, пароля и ссылкой на авторизацию уже зарегистрированных пользователей.

Профиль:

HTML:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Профиль</title>

    <link rel="stylesheet" href="{{ url_for('static', path =
'/style_profile.css') }}">

    <!--Подключаю шрифты Awesome-->

    <script src="https://kit.fontawesome.com/758af4b3e3.js"
crossorigin="anonymous"></script>

</head>

<body>

    <!-------Кнопка луны, тёмная тема----->

    <div class="dark-light" id="themeToggle">

        <i class="fa-regular fa-moon"></i>

    </div>


    <div class="container">

        <div class="profile-box">

            <a href="/"></a>

            <a href="/settings"></a>

            <h3>{{user.username}}</h3>
```

```

        <p>Студент ОмГПУ</p>

    </div>

</div>

    <script src="{ { url_for('static', path = '/DarkAndWhite.js')
}}"></script>

</body>

</html>

```

CSS:

```

/*profile*/

*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: "poppins", sans-serif;

}

/*-----Тёмная и светла тема-----*/

body.dark-theme {

    --background: #0b1114;

}

body.light-theme {

    --background: rgb(221, 221, 221);

}

body {

    background-color: var(--background);

}

.container{

    width: 100%;

```



```
height: 92vh;

background-color: var(--background);

background-size: cover;

background-position: center;

display: flex;

align-items: center;

justify-content: center;

}
```

```
.profile-box{

background: #9508CD;

text-align: center;

padding: 40px 90px;

color: #F8F4FA;

position: relative;

border-radius: 20px;

}
```

```
.menu-icon{

width: 25px;

position: absolute;

left: 40px;

top: 40px;

transition: all 0.3s ease-in-out;

}
```

```
.menu-icon:hover{

transform: scale(1.2);

color: #F8F4FA;

}
```

```
.setting-icon{
```

```
width: 25px;

position: absolute;

right: 40px;

top: 40px;

transition: all 0.3s ease-in-out;
}

.setting-icon:hover{

transform: scale(1.2);

color: #F8F4FA;
}

.ava-icon{

height: 200px;

width: 200px;

border-radius: 50%;

background: #F8F4FA;

padding: 6px;

margin-top: 40px;
}

.icons{

display: flex;

justify-content: space-around;

align-items: center;
}

.image{

margin: 15px 10px;

transition: all 0.3s ease-in-out;
}
```

```

.image:hover{

    transform: scale(1.2);

    color: #F8F4FA;

}

.profile-box h3 {

    font-size: 22px;

    margin-top: 20px;

    font-weight: 500;

}

.a {

    display: block;

    width: 100%;

    height: 100%;

}

/*-----Луна справа сверху-----
*/

.dark-light {

    color: #9508CD;

    margin-left: 97%;

    margin-top: 1.7%;

}

.dark-light i {

    font-size: 28px;

    margin: 0;

    cursor: pointer;

}

```




```

@app.get("/profile", response_class=HTMLResponse)

def profile(request: Request, db: Session =
Depends(database.get_db), usr_id: Union[str, None] =
Cookie(default=None)):

    if usr_id:

        user = db.query(models.User).filter(models.User.id ==
usr_id).first()

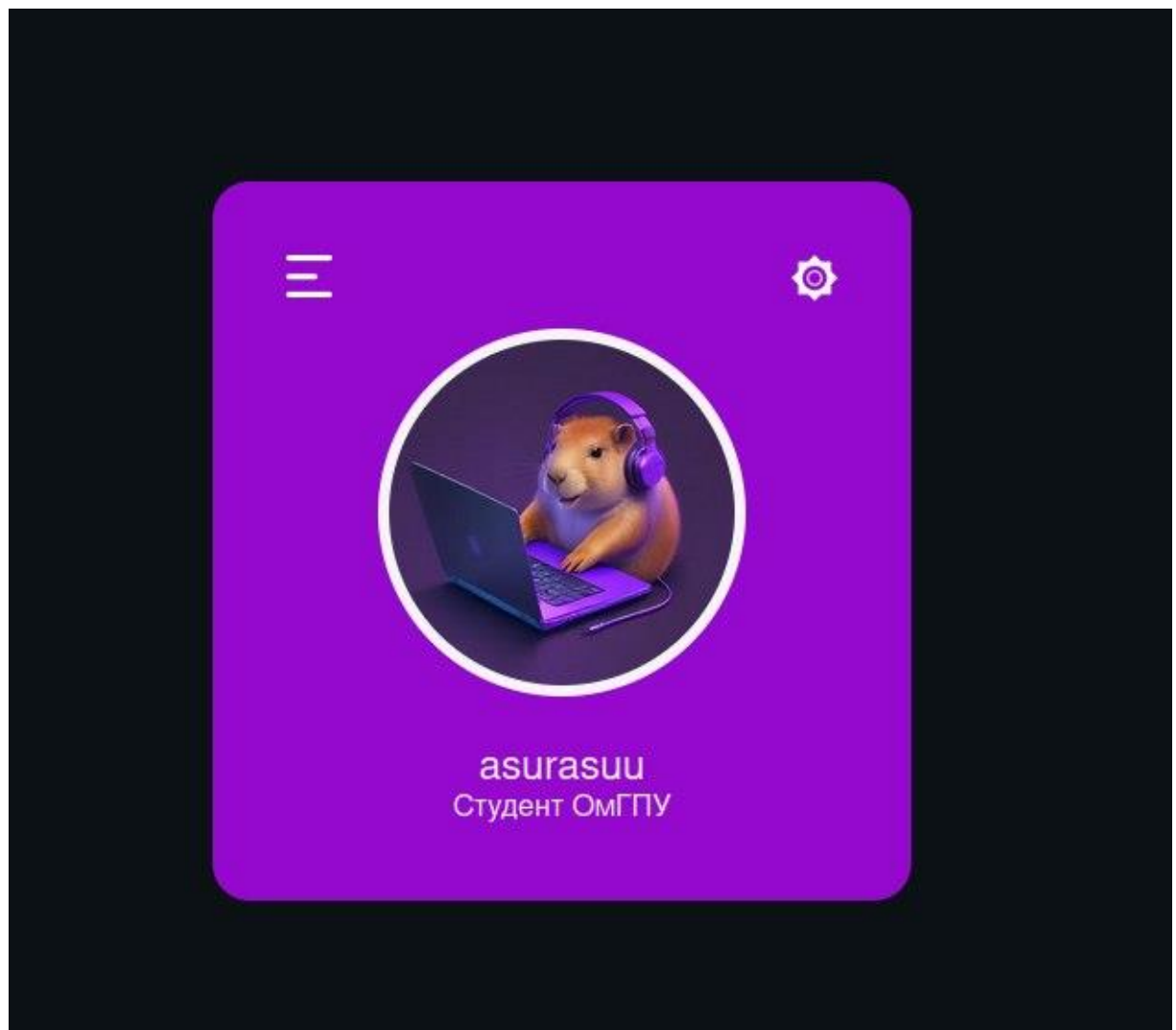
        template = "profile.html"

        context = {"request": request, 'user': user}

        return templates.TemplateResponse(template, context)

    return RedirectResponse("/auth")

```



Настройка профиля:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Профиль</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://kit.fontawesome.com/31b0575cc7.js"
crossorigin="anonymous" referrerpolicy="no-referrer"></script>

</head>
<body>
  <div class="container">
    <div class="profile">
      <div class="profile-header">
        
        <div class="profile-text-container">
          <h1 class="profile-title">Профиль</h1>
          <p class="profile-email">
            alextorpv@gmail.com
          </p>
        </div>
      </div>
    </div>

    <div class="menu">
      <a href="#" class="menu-link"><i class="fa-solid fa-user-tie
menu-icon"></i>Основные</a>
      <a href="#" class="menu-link"><i class="fa-solid fa-key menu-
icon"></i>Пароль</a>
      <a href="#" class="menu-link"><i class="fa-solid fa-address-card
menu-icon"></i>Профиль</a>
      <a href="#" class="menu-link"><i class="fa-solid fa-right-from-
bracket menu-icon"></i>Выйти</a>
    </div>
  </div>

  <form class="account">
    <div class="account-header">
      <h1 class="account-title">Основное</h1>
      <div class="btn-container">
        <button class="btn-ok">Сохранить</button>
        <button class="btn-cancel">Отмена</button>
      </div>
    </div>

    <div class="account-edit">
      <div class="input-container">
        <label>Фамилия</label>
        <input type="text" placeholder="Фамилия">
      </div>
      <div class="input-container">
        <label>Имя</label>
        <input type="text" placeholder="Имя">
      </div>
    </div>
  </form>
</body>
```

```

        <div class="account-edit">
            <div class="input-container">
                <label>Email</label>
                <input type="text" placeholder="Email">
            </div>
            <div class="input-container">
                <label>Номер телефона</label>
                <input type="text" placeholder="Номер телефона">
            </div>
        </div>

        <div class="account-edit">
            <div class="input-container">
                <label>0 себе</label>
                <input type="text" placeholder="0 себе">
            </div>
        </div>

        <div class="account-edit">
            <div class="input-container2">
                <label>Страна</label>
                <select class="custom-select">
                    <option>Казахстан</option>
                    <option selected>Россия</option>
                    <option>Республика Беларусь</option>
                    <option>Китай</option>
                </select>
            </div>
        </div>
    </form>
</div>
</body>
</html>

```

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

body {
    background-color: #0b1114;;
}

.container {
    display: flex;
    margin: 50px 10px;
}

.profile {
    flex: 1;
    margin-right: 10px;
    background-color: #262626;
    box-shadow: 0px 1px 4px rgba(0, 0, 0, 0.25);
    border-radius: 5px;
    height: max-content;
}

```

```

.profile-header {
  display: flex;
  margin-left: 15px;
}

.profile img {
  width: 60px;
  height: 60px;
  margin: 10px;
}

.profile-text-container {
  line-height: 0.5;
}

.profile-title {
  font-family: "Poppins", sans-serif;
  font-size: 20px;
  color: aliceblue;
}

.profile-email {
  font-family: "Poppins", sans-serif;
  font-size: 14px;
  color: #fff;
}

.menu {
  margin: 0 20px;
}

.menu-link {
  display: block;
  text-decoration: none;
  color: aliceblue;
  font-family: "Poppins", sans-serif;
  padding: 10px;
  margin: 10px;
  border-radius: 10px;
}

.menu-icon {
  margin-right: 10px;
}

.menu-link:first-child {
  color: aliceblue;
  background-color: #9508CD;
}

.menu-link:hover {
  background-color: #5d6061;
  color: #fff;
}

.menu-link:first-child:hover {
  background-color: #9508CD;
  color: #fff;
}

```

```

}

.account {
  flex: 2;
  margin-left: 10px;
  background-color: #262626;
  box-shadow: 0px 1px 4px rgba(0, 0, 0, 0.25);
  border-radius: 5px;
  height: max-content;
}

.account-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px;
  border-bottom: 1px solid #fff;
}

.account-title {
  font-family: "Poppins", sans-serif;
  font-size: 20px;
  color: aliceblue;
  font-weight: 500;
  margin-left: 10px;
}

.btn-container {
  margin-right: 5px;
}

.btn-ok {
  width: 8em;
  height: 3em;
  cursor: pointer;
  border: 1px solid #0b1114;
  border-radius: 5px;
  color: #fff;
  font-weight: bold;
  margin-right: 6px;
  background-color: #9508CD;
  transition-duration: 0.3s;
  box-shadow: rgba(0, 0, 0, 0.25);
}

.btn-ok:hover {
  background-color: #5d6061;
}

.btn-cancel {
  width: 8em;
  height: 3em;
  cursor: pointer;
  border: 1px solid #0b1114;
  border-radius: 5px;
  color: #fff;
  font-weight: bold;
  margin-right: 6px;
  background-color: #768499;
}

```

```

    transition-duration: 0.3s;
    box-shadow: rgba(0, 0, 0, 0.25);
}

.btn-cancel:hover {
    background-color: #5d6061;
}

.account-edit {
    display: flex;
    justify-content: space-between;
    margin: 15px 0;
}

.input-container {
    width: 100%;
    display: flex;
    flex-direction: column;
    margin: 0 20px;
}

.input-container label {
    font-family: "Poppins", sans-serif;
    color: #fff;
    font-size: 14px;
}

.input-container input {
    height: 25px;
    border: 1.5px solid #5d6061;
    border-radius: 5px;
    padding: 5px;
    color: #5d6061;
    font-family: "Poppins", sans-serif;
}

.input-container input:focus {
    outline: none;
    border: 1.5px solid #9508CD;
}

.input-container textarea {
    display: block;
    width: 100%;
    height: 100px;
    padding: 5px;
    border: 1.5px solid #5d6061;
    border-radius: 5px;
    font-family: "Poppins", sans-serif;
    color: #5d6061;
}

.input-container textarea:focus {
    outline: none;
    border: 1.5px solid #9508CD;
}

.input-container2 label {
    font-family: "Poppins", sans-serif;

```



```

    color: #fff;
    font-size: 14px;
}

.input-container2 {
    width: 100%;
    display: flex;
    flex-direction: column;
    margin: 0 20px;
}

.input-container2 input {
    height: 25px;
    border: 1.5px solid #5d6061;
    border-radius: 5px;
    padding: 5px;
    color: #5d6061;
    font-family: "Poppins", sans-serif;
}

.input-container2 input:focus {
    outline: none;
    border: 1.5px solid #9508CD;
}

.input-container2 textarea {
    display: block;
    width: 100%;
    height: 100px;
    padding: 5px;
    border: 1.5px solid #5d6061;
    border-radius: 5px;
    font-family: "Poppins", sans-serif;
    color: #5d6061;
}

.input-container2 textarea:focus {
    outline: none;
    border: 1.5px solid #9508CD;
}

@media screen and (max-width: 768px) {
    .container {
        flex-direction: column;
    }

    .profile {
        margin-bottom: 20px;
        margin-right: 0;
    }

    .account {
        margin: 0;
    }

    .account-header {
        flex-direction: column;
    }
}

```

```

.account-edit {
    flex-direction: column;
    margin: 10px;
}

.input-container {
    margin: 10px;
}

.input-container input {
    margin-right: 10px;
}

.input-container textarea {
    display: block;
    width: auto;
    margin-right: 10px;
}

.input-container2 {
    margin: 10px;
}

.input-container2 input {
    margin-right: 10px;
}

.input-container2 textarea {
    display: block;
    width: auto;
    margin-right: 10px;
}
}

```

PYTHON

```

@app.get("/settings", response_class=HTMLResponse)

def read_item(request: Request, db: Session =
Depends(database.get_db), usr_id: Union[str, None] =
Cookie(default=None)):

    user = db.query(models.User).filter(models.User.id ==
usr_id).first()

    if usr_id is not None:

        template = "settings.html"

```

```

context = {"request": request, 'user': user}

return templates.TemplateResponse(template, context)

return RedirectResponse("/auth")

```

```

@app.post("/saveSettings")

def save_settings(request: Request, surname: str = Form(...), name:
str = Form(...), email: str = Form(...),

phone: str = Form(...), about: str = Form(...),

db: Session = Depends(database.get_db),

usr_id: Union[str, None] = Cookie(default=None)):

    if usr_id is not None:

        user = db.query(models.User).filter(models.User.id ==
usr_id).update({

            "name": name, "surname": surname,

            "phone": phone, "email": email,

            "about": about

        })

        db.commit()

        try:

            db.refresh(user)

        except UnmappedInstanceError as e:

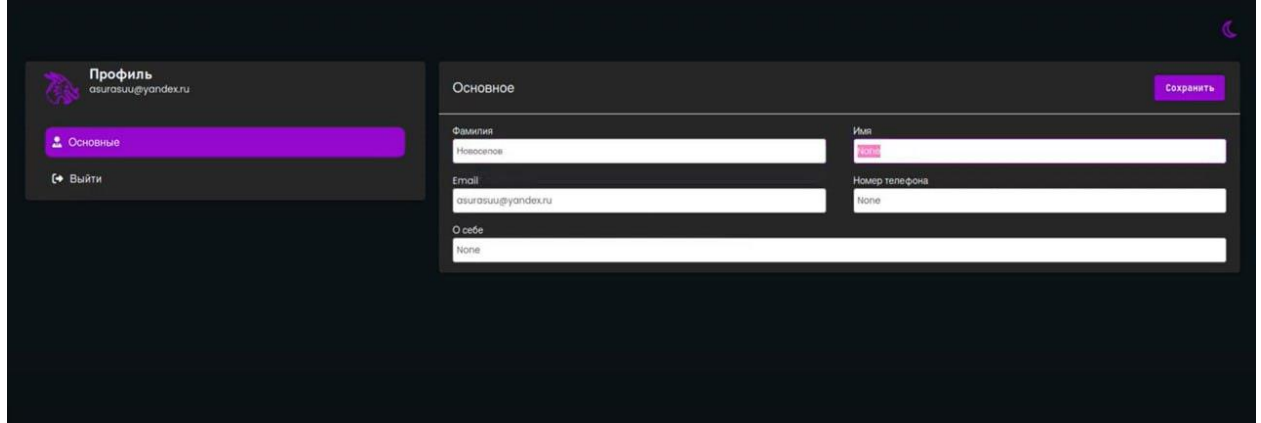
            pass

        resp = RedirectResponse("/settings", status_code = 302)

        return resp

    return RedirectResponse("/auth")

```



В результате, страница включает информацию о профиле пользователя, меню навигации и форму для редактирования данных аккаунта. Форма содержит поля для фамилии, имени, email, номера телефона и информации о себе. Подключается внешний файл стилей и библиотека иконок Font Awesome для дополнительной стилизации.

Мессенджер:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Мессенджер</title>

    <link rel="stylesheet" href="{{ url_for('static', path =
'style_messenger.css') }}">

    <!--Подключаю шрифты Awesome-->

    <script src="https://kit.fontawesome.com/758af4b3e3.js"
crossorigin="anonymous"></script>

</head>

<body>

    <!-------Панель навигации----->

    <div class="oneColumn">

        <div class="Logo">

            </img>

            <a class="A">NEON</a>

            <a class="B">inst</a>

        </div>

        <div class="Navigation">

            <i class="fa-regular fa-user"></i>

            <a href="/profile">Профиль</a>
```

```

</div>

<div class="Navigation">
  <i class="fa-regular fa-message"></i>
  <a href="/messenger">Мессенджер</a>
</div>

<div class="Navigation">
  <i class="fa-regular fa-compass"></i>
  <a href="/">Лента</a>
</div>

<div class="Navigation">
  <i class="fa-solid fa-people-group"></i>
  <a href="/about">0 проекте</a>
</div>

<div class="Navigation">
  <i class="fa-solid fa-gear"></i>
  <a href="/settings">Настройки</a>
</div>

</div>

</div>

<div class="app">
  <div class="header">

    <div class="dark-light" id="themeToggle">
      <i class="fa-regular fa-moon"></i>
    </div>

```

```

        <!-- <div class="search">

            <form>

                <input type="text" placeholder="Искать здесь...">

                <button type="submit"></button>

            </form>

        </div> -->

        <div class="user-settings">

        </div>

    </div>

    <!-- Доступные чаты -->

    <div class="wrapper">

        <div class="conversation-area">

            {% for user in users %}

                <a href="/messenger/{{ user.id }}">

                    <div class="msg">

                        <div class="msg-detail">

                            <div class="msg-username">{{ user.username
}}</div>

                            <div class="msg-content">

                                <!-- <span class="msg-message">Великий cs-
сер</span>

                                <span class="msg-date">1day</span> -->

                            </div>

                        </div>

                    </div>

                </a>

            {% endfor %}

        </div>

```

```

<!--Мессенджер-->

{% if iid %}

    <div class="chat-area-main">

        <h2 style="text-align: center; padding-bottom: 10px;">Чат
c {{ iid.username }}</h2>

        <div class="messages-all-block" style="height: 650px;
overflow-y: scroll;">

            {% for message in messages %}

                {% if message.sender_id == user.id %}

                    <div class="chat-msg owner">

                        <div class="chat-msg-profile">

                            <!-- <div class="chat-msg-
username">Lonni</div> -->

                            <!-- <div class="chat-msg-date">Message
seen 1.22pm</div> -->

                        </div>

                        <div class="chat-msg-content">

                            <div class="chat-msg-text">{{
message.content }}</div>

                        </div>

                    </div>

                {% else %}

                    <div class="chat-msg">

                        <div class="chat-msg-profile">

                            <!-- <div class="chat-msg-
username">Lonni</div> -->

                            <!-- <div class="chat-msg-date">Message
seen 1.22pm</div> -->

                        </div>

                        <div class="chat-msg-content">

```



```

                                <div class="chat-msg-text">{{
message.content }}</div>

                                </div>

                                </div>

                                {% endif %}

                                {% endfor %}

                                </div>

                                <form action="/sendmessage" method="post">

                                    <div class="chat-area-footer">

                                        <button class="transparent-button"><i class="fa"
style="font-size:24px; display: none">&#xf0c6;</i></button>

                                        <input type="text" name="from_user_id" value="{{
iid.id }}" style="display: none">

                                        <input type="text" name="text_message"
placeholder="Напиши сообщение..." autofocus required />

                                        <button class="transparent-button"><i class="fa"
style="font-size:24px">&#xf1b0;</i></button>

                                    </div>

                                </form>

                                </div>

                                </div>

                                {% endif%}

                                </div>

                                <script src="{{ url_for('static', path = '/DarkAndWhite.js')
}}"></script>

                                </body>

                                </html>

```

CSS:

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300
;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,
800;1,900&display=swap');
```

```
body.dark-theme {

  --body-font: "Manrope", sans-serif;

  --color-purple: #9508CD;

  --color-black: #0b1114;

  --color-white: white;

  background-color: #0b1114;

  --border-color: white;

}
```

```
body.light-theme {

  --body-font: "Manrope", sans-serif;

  --color-purple: #9508CD;

  --color-black: white;

  --color-white: black;

  background-color: white;

  --border-color: black;

}
```

```
body{

  font-family: var(--body-font);

  background-color: var(--color-black);

  --body-font: "Manrope", sans-serif;

  --color-purple: #9508CD;

  --color-white: white;
```

```
--color-black: #0b1114;

--border-color: white;

}


*{

    margin: 0%;

    font-family: "Poppins", sans-serif;

    outline: none;

    box-sizing: border-box;

}


a {

    font-size: 14px;

    margin: 15%;

    text-decoration: none;

    color: var(--color-white);

}


body {

    background-color: var(--color-black);

    font-family: var(--body-font);

    color: var(--color-white);

}


.app {

    position: absolute;

    top: 0;

    left: 20%;
```

```

        width: 80%;

        height: 100vh;

        overflow-y: auto;

        scrollbar-width: none;
    }

    /*-----Logo-----*/

    .Logo {

        margin-left: 5%;

        display: flex;

        align-items: center;

        margin-bottom: 17%;

    }

    .Logo img {

        margin: 0;

    }

    .A {

        margin-left: 2%;

        margin-right: 0%;

        font-size: 42px;

        color: var(--color-white);

    }

    .B {

        margin-left: 0%;

        font-size: 42px;

        color: var(--color-purple);

    }

    /*-----Панель нафигации слева-----
    */

```

```
.Navigation {  
    margin-left: 10%;  
    display: flex;  
    align-items: center;  
}  
  
.Navigation a {  
    margin: 7%;  
    font-size: 16px;  
    color: var(--color-white);  
}  
  
.Navigation i {  
    color: var(--color-purple);  
    font-size: 30px;  
}
```

```
/*-----первая колонка-----*/  
-----*/
```

```
.oneColumn{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 20%;  
    height: 100vh;  
    background-color: var(--color-black);  
}
```

```
.header {  
  
    height: 80px;  
  
    width: 100%;  
  
    border-bottom: 1px solid var(--border-color);  
  
    border-color: var(--border-color); /*одна из полос*/  
  
    display: flex;  
  
    align-items: center;  
  
    padding: 0 20px;  
  
}
```

```
.wrapper {  
  
    width: 100%;  
  
    height: 89%;  
  
    display: flex;  
  
    flex-grow: 1;  
  
    overflow: hidden;  
  
    border-right: 1px solid var(--border-color);  
  
}
```

```
.conversation-area,  
.detail-area {  
  
    width: 340px;  
  
    flex-shrink: 0;  
  
}
```

```
.detail-area {  
  
    border-left: 1px solid var(--border-color);  
  
    margin-left: auto;  
  
    padding: 30px 30px 0 30px;  
  
    display: flex;  
  
    flex-direction: column;  
  
    overflow: auto;
```

```

}

.chat-area {
  flex-grow: 1;
}

/*-----поле ввода сообщений-----*/

input {
  height: 100%;
  width: 100%;
  display: block;
  background-color: transparent;
  border: none;
  color: var(--color-white);
  padding: 0;
  background-repeat: no-repeat;
  background-size: 16px;
  background-position: 25px 48%;
  font-family: var(--body-font);
  font-weight: 600;
  font-size: 15px;
}

/*-----Луна справа сверху-----
*/

.dark-light {
  position: fixed;
  align-self: center;
  margin-left: 75%;

```

```
color: var(--color-purple);
```

```
flex-shrink: 0;
```

```
}
```

```
.dark-light i {
```

```
font-size: 28px;
```

```
}
```

```
/*-----Настройка размера изображений-----  
-----*/
```

```
.user-profile {
```

```
width: 40px;
```

```
height: 40px;
```

```
border-radius: 50%;
```

```
}
```

```
.msg-profile {
```

```
width: 44px;
```

```
height: 44px;
```

```
border-radius: 50%;
```

```
object-fit: cover;
```

```
margin-right: 15px;
```

```
}
```

```
/*-----Панель с пользователями-----*/
```

```
.conversation-area {
```

```
border-right: 1px solid var(--border-color);
```

```
border-color: var(--border-color);/*одна из полос*/
```

```
overflow-y: auto;
```

```
overflow-x: hidden;
```

```
display: flex;
```



```
flex-direction: column;

position: relative;

}
```

```
.msg {

display: flex;

align-items: center;

padding: 20px;

cursor: pointer;

transition: 0.2s;

position: relative;

}
```

```
.msg-username {

margin-bottom: 4px;

font-weight: 600;

font-size: 15px;

}
```

```
.msg-message {

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

color: var(--msg-message);

}
```

```
.msg-date {
```

```

    font-size: 14px;

    color: var(--msg-date);

    margin-left: 3px;
}

/*-----Кнопка добавить-----*/
-----*/

.add {

    position: sticky;

    bottom: 25px;

    background-color: var(--color-purple);

    width: 60px;

    height: 60px;

    border: 0;

    border-radius: 50%;

    background-image: url("data:image/svg+xml;charset=UTF-8,%3csvg
xmlns='http://www.w3.org/2000/svg' width='24' height='24' fill='none'
stroke='white' stroke-width='2' stroke-linecap='round' stroke-linejoin='round'
class='feather feather-plus'%3e%3cpath d='M12 5v14M5 12h14'/%3e%3c/svg%3e");

    background-repeat: no-repeat;

    background-position: 50%;

    background-size: 28px;

    box-shadow: 0 0 16px var(--color-purple);

    margin: auto auto -55px;

    flex-shrink: 0;

    z-index: 1;

    cursor: pointer;
}

.overlay {

    position: sticky;

    bottom: 0;

```

```

    left: 0;

    width: 340px;

    height: 80px;
}

/*-----OKHO-----*/

.chat-area-main {
    min-width: 600px;
    max-width: 100%;
    height: 100vh;
    padding-top: 15px;
}

.chat-msg-img {
    height: 40px;
    width: 40px;
    border-radius: 50%;
    object-fit: cover;
}

.chat-msg-profile {
    flex-shrink: 0;
    margin-top: auto;
    margin-bottom: -20px;
    position: relative;
}

.chat-msg-date {
    position: absolute;

```

```

    left: calc(100% + 12px);

    bottom: 0;

    font-size: 12px;

    font-weight: 600;

    color: var(--color-white);

    white-space: nowrap;
}

.chat-msg {
    display: flex;

    padding: 0 20px 45px;

    &-content {
        margin-left: 12px;

        max-width: 70%;

        display: flex;

        flex-direction: column;

        align-items: flex-start;
    }

    &-text {
        background-color: var(--color-black);

        padding: 15px;

        border-radius: 20px 20px 20px 0;

        line-height: 1.5;

        font-size: 14px;

        font-weight: 500;

        & + & {
            margin-top: 10px;
        }
    }
}

.chat-msg-text {

```

```

    color: var(--color-white);

    padding-left: 10px;
}

.owner {
    flex-direction: row-reverse;

    .chat-msg-content {
        margin-left: 0;
        margin-right: 12px;
        align-items: flex-end;
    }
}

/*-----Поле ввода сообщений-----*/

.chat-msg-date {
    left: auto;
    right: calc(100% + 12px);
}
}

.chat-msg-text img {
    max-width: 300px;
    width: 100%;
}

.chat-area-footer {
    /* margin-top: 10px; */
    display: flex;
    border-top: 1px solid var(--color-purple);
    width: 100%;
    padding: 10px 20px;
    align-items: center;
    background-color: var(--color-black);
}

```

```
    position: sticky;

    bottom: 0;

    left: 0;

    border-radius: 1px;
}
```

```
.chat-area-footer input {

    border: none;

    color: var(--body-color);

    background-color: var(--input-bg);

    padding: 12px;

    border-radius: 6px;

    font-size: 15px;

    margin: 0 12px;

    width: 100%;
}
```

```
.detail-area-header {

    display: flex;

    flex-direction: column;

    align-items: center;
}
```

```
/*-----Окно поиска-----
--*/

.search {

    padding-top: 1%;

    position: relative;

    padding-left: 40%;
}
```

```
}

.search input {
width: 400px;
height: 44px;
padding-left: 10px;
border: 2px solid var(--color-purple);
border-radius: 5px;
outline: none;
background: var(--color-black);
color: var(--color-white);
}
```

```
.search button {
position: absolute;
top: 0;
width: 20%;
height: 100%;
right: -80px;
border: none;
background: var(--color-purple);
border-radius: 0 5px 5px 0;
cursor: pointer;
}
```

```
.search button:before {
content: "\f002";
font-family: FontAwesome;
font-size: 16px;
color: var(--color-white);
}
```

```
.search form {
```

```

    position: relative;

    width: 80%;

    margin: auto;
}

/*-----Кнопки возле отправки сообщения-----*/

.transparent-button {

    background: none;

    border: none;

    cursor: pointer;

    color: var(--color-purple);
}

```

PYTHON

```

# Страница сообщений

@app.get("/messenger/", response_class=HTMLResponse)

def read_item(request: Request, db: Session =
Depends(database.get_db), usr_id: Union[str, None] =
Cookie(default=None)):

    if usr_id is not None:

        user = db.query(models.User).filter(models.User.id ==
usr_id).first()

        users = db.query(models.User).filter(models.User.id !=
usr_id).all()

        template = "messenger.html"

        context = {"request": request, 'users': users, 'user': user,
'iid': None}

        return templates.TemplateResponse(template, context)

```



```

        return RedirectResponse("/auth") # Перенаправление на страницу
входа

@app.get("/messenger/{id}", response_class=HTMLResponse)

def read_item(request: Request, id: int, db: Session =
Depends(database.get_db), usr_id: Union[str, None] =
Cookie(default=None)):

    if usr_id is not None:

        user = db.query(models.User).filter(models.User.id ==
usr_id).first()

        users = db.query(models.User).filter(models.User.id !=
usr_id).all()

        # Модель собеседника

        iid = db.query(models.User).filter(models.User.id ==
id).first()

        # Все сообщения

        messages = db.query(models.Message).filter(

            or_(

                and_(models.Message.sender_id == usr_id,
models.Message.recipient_id == iid.id),

                and_(models.Message.sender_id == iid.id,
models.Message.recipient_id == usr_id)

            )

        ).all()

        template = "messenger.html"

        context = {"request": request, 'users': users, 'user': user,
'iid': iid, 'messages': messages}

        return templates.TemplateResponse(template, context)

    return RedirectResponse("/auth") # Перенаправление на страницу
входа

```

```

@app.post("/sendmessage")

def send_message(from_user_id: str = Form(...), text_message: str =
Form(...), db: Session = Depends(database.get_db), usr_id: Union[str,
None] = Cookie(default=None)):

    if usr_id is not None:

        new_message = models.Message(sender_id=usr_id,
recipient_id=from_user_id, content=text_message)

        db.add(new_message)

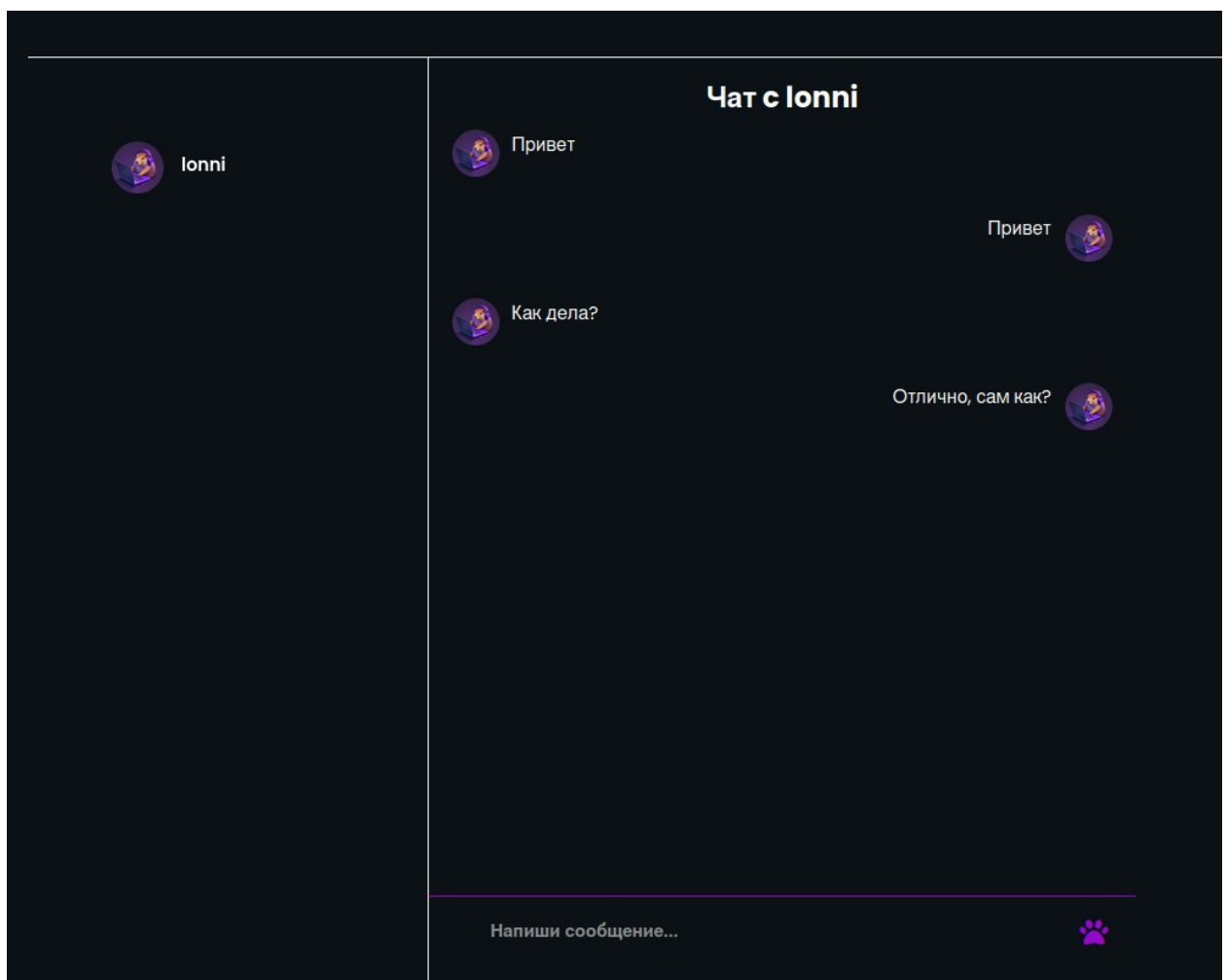
        db.commit()

        db.refresh(new_message)

    return RedirectResponse("/messenger/" + from_user_id,
status_code=302)

return RedirectResponse("/auth") # Перенаправление на страницу
входа

```



В результате, данная страница позволяет пользователям мгновенно обмениваться сообщениями с другими пользователями социальной сети NeonInst

HTML:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Neoinst</title>

    <!--Подключаю стиль css-->

    <!-- style/index.css -->

    <link rel="stylesheet" href="{{ url_for('static',
path='/index.css') }}">

    <!--Подключаю шрифты Awesome-->

    <script src="https://kit.fontawesome.com/758af4b3e3.js"
crossorigin="anonymous"></script>

<style>

    /* Стиль для затемнения фона */

    .modal-overlay {

        display: none;

        position: fixed;

        top: 0;

        left: 0;

        width: 100%;

        height: 100%;

        background: rgba(0, 0, 0, 0.5);

        z-index: 1;

    }

    /* Стиль для модального окна */

    .modal-content {
```

```

        display: none;

        position: fixed;

        top: 50%;

        left: 50%;

        transform: translate(-50%, -50%);

        background: white;

        padding: 20px;

        box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);

        z-index: 2;

        border-radius: 10px;

    }

    /* Кнопка закрытия модального окна */

    .close-button {

        position: absolute;

        top: 10px;

        right: 10px;

        background: #9508CD;

        color: white;

        border: none;

        padding: 5px 10px;

        cursor: pointer;

    }

</style>

</head>

<body>

    <!--Панель навигации-->

    <div class="oneColumn">

        <div class="Logo">

```

```

</img>

<a class="A">NEON</a>

<a class="B">inst</a>

</div>

<div class="Navigation">

    <i class="fa-regular fa-user"></i>

    <a href="/profile">Профиль</a>

</div>

<div class="Navigation">

    <i class="fa-regular fa-message"></i>

    <a href="/messenger">Мессенджер</a>

</div>

<div class="Navigation">

    <i class="fa-regular fa-compass"></i>

    <a href="/">Лента</a>

</div>

<div class="Navigation">

    <i class="fa-solid fa-people-group"></i>

    <a href="/about">О проекте</a>

</div>

<div class="Navigation">

    <i class="fa-solid fa-gear"></i>

    <a href="/settings">Настройки</a>

</div>

</div>

```

```

</div>

<div class='twoColumn'>

    <div class="search">

        <!-- <form>

            <input type="text" placeholder="Искать здесь...">

            <button type="submit"></button>

        </form> -->

        <div class="twoColumnDaughter">

            {% for photo in photos%}

                <!-- <div class="post_block">

                    <div class="post_header">

                        <span>{{user.username}}</span>

                    </div>

                    <div class="post_img">

                    </div>

                    <div class="post_footer">

                        <p>{{ photo.text }}</p>

                    </div>

                </div> -->

                <!-- <div class="services-list"> -->

                <div class="card">

                    <div class="img1">

```

```

        <h2>{{user.username}}</h2>

    </div>

    <div class="img2-container">

    </div>

    <div class="text-container">

        <p>

            {{ photo.text }}

        </p>

    </div>

</div>

<!-- </div> -->

    {% endfor %}

</div>

</div>

</div>

<div class='threeColumn'>

    <!-------Кнопка луны, тёмная тема----->

    <div class="dark-light" id="themeToggle">

        <i class="fa-regular fa-moon"></i>

    </div>

    <!-------Прогноз погоды----->

    <div class="threeColumnDaughter" id="ww_9a080cb11bca2" v='1.3'
loc='id'
a='{"t":"horizontal","lang":"ru","ids":["wl4528"],"font":"Arial","sl_ics":
"one_a","sl_sot":"celsius","cl_bkg":"#7B1FA2","cl_font":"#FFFFFF","cl_clou

```



```

d": "#FFFFFF", "cl_persp": "#81D4FA", "cl_sun": "#FFC107", "cl_moon": "#FFC107", "
cl_thund": "#FF5722"}'>

    <a href="https://weatherwidget.org/" id="ww_9a080cb11bca2_u"
target="_blank">Html weather widget</a>

</div>

<script async
src="https://app2.weatherwidget.org/js/?id=ww_9a080cb11bca2"></script>

<!-------Окно публикация----->
----->

<div class="pub">

    <button id="uploadBtn">

        <a>Публикация</a>

        <a>#xf055;</a>

    </button>

</div>

<!-- Затемняющий фон -->

<div id="modalOverlay" class="modal-overlay"></div>

<!-- Модальное окно -->

<div id="modalContent" class="modal-content">

    <button class="close-button" id="closeBtn">X</button>

    <form id="uploadForm">

        <label for="file">Выберите фото:</label>

        <input type="file" id="file" name="file" accept="image/*"
required>

        <br><br>

        <label for="user">Подпись:</label>

        <input type="text" id="user" name="text" required>

        <button type="submit">Загрузить</button>

    </form>

</div>

```

```

    <script src="{{ url_for('static', path='/DarkAndWhite.js')
}}"></script>

<script>

    const uploadBtn = document.getElementById("uploadBtn");

    const modalOverlay = document.getElementById("modalOverlay");

    const modalContent = document.getElementById("modalContent");

    const closeBtn = document.getElementById("closeBtn");

    uploadBtn.onclick = function() {

        modalOverlay.style.display = "block";

        modalContent.style.display = "block";

    }

    closeBtn.onclick = function() {

        modalOverlay.style.display = "none";

        modalContent.style.display = "none";

    }

    modalOverlay.onclick = function() {

        modalOverlay.style.display = "none";

        modalContent.style.display = "none";

    }

    document.getElementById("uploadForm").onsubmit = async
function(event) {

        event.preventDefault();

        const formData = new FormData(this);

        const response = await fetch('/uploadfiles/', {

            method: 'POST',

            body: formData

```

```

    });

    modalOverlay.style.display = "none";
    modalContent.style.display = "none";

    if( response ) {
        window.location.reload(true);
    }
}

</script>

</body>

</html>

```

CSS:

```

/*общие настройки файла index.html*/

@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

/*-----Тёмная и светла тема-----*/

body.dark-theme {

    --body-font: "Manrope", sans-serif;

    --color-purple: #9508CD;

    --color-black: #0b1114;

    --color-white: white;

    background-color: #0b1114;

}

body.light-theme {

    --body-font: "Manrope", sans-serif;

```

```

--color-purple: #9508CD;

--color-black: white;

--color-white: black;

background-color: white;
}


body{

--body-font: "Manrope", sans-serif;

--color-purple: #9508CD;

--color-black: #0b1114;

--color-white: white;

--color-threeCol: #0b1114;

--color-moon: #9508CD;

background-color: #0b1114;


font-family: var(--body-font);

transition: background-color 0.3s ease;

}


*{

margin: 0%;

font-family: "Poppins", sans-serif;

}


p{

margin: 15%;

font-size: 14px;

font-weight: bold;

color: var(--color-white);

font-family: var(--body-font);

}

```

```

a {
    font-size: 14px;
    margin: 15%;
    text-decoration: none;
    color: var(--color-white);
}

/*-----Logo-----
--*/

.Logo {
    margin-left: 5%;
    display: flex;
    align-items: center;
    margin-bottom: 17%;
}

.Logo img {
    margin: 0;
}

.A {
    margin-left: 2%;
    margin-right: 0%;
    font-size: 42px;
    color: var(--color-white);
}

.B {
    margin-left: 0%;
    font-size: 42px;
    color: var(--color-purple);
}

```

```

/*-----Панель нафигации слева-----
-----*/

.Navigation {
    margin-left: 10%;

    display: flex;

    align-items: center;
}

.Navigation a {
    margin: 7%;

    font-size: 16px;

    color: var(--color-white);
}

.Navigation i {
    color: var(--color-purple);

    font-size: 30px;
}

/*-----первая колонка-----
-----*/

.oneColumn{
    position: absolute;

    top: 0;

    left: 0;

    width: 20%;

    height: 100vh;

    background-color: var(--color-black);
}

/*-----вторая колонка-----
-----*/

.twoColumn{

```

```

background-color: var(--color-black);

position: absolute;

top: 0;

left: 20%;

width: 50%;

height: 100vh;

overflow-y: auto;

scrollbar-width: none;

}

.twoColumnDaughter{

/* border: 1px solid yellow; */

-webkit-border-radius: 15px;

border-radius: 15px;

background-color: var(--color-black);

padding: 0.1%;

margin: 4%;

/* text-align: center; */

display: flex;

justify-content: center;

flex-wrap: wrap;

}

.card {

background: #262626;

padding: 40px;

font-size: 13px;

font-weight: 300;

border-right: 8px;

transition: background 0.5s, transform 0.5s;

border-radius: 15px;

display: flex;

```

```
flex-direction: column;

align-items: center;

margin-bottom: 10px;
}

.img1 {

align-self: start;

display: flex;

align-items: center;
}

.img1 img {

margin: 3%;

max-width: 50px;

border-radius: 50%;
}

.img1 h2 {

display: block;

margin-left: 2%;

margin-right: 0%;

font-size: 20px;

color: #F8F4FA;
}

.img2-container {

margin-top: 20px;

display: flex;

justify-content: center;
}

.img2 {

max-width: 400px;
```



```
}
```

```
.text-container {  
    margin-top: 20px;  
    text-align: center;  
}
```

```
.text-container p {  
    font-size: 16px;  
    color: #F8F4FA;  
}
```

```
/*-----третья колонка-----  
-----*/
```

```
.threeColumn{  
    position: absolute;  
    top: 0;  
    right: 0;  
    width: 30%;  
    background-color: var(--color-black);  
    margin-top: 1.7%;  
}
```

```
.threeColumnDaughter {  
    -webkit-border-radius: 15px;  
    border-radius: 15px;  
    background-color: var(--color-black);  
}
```

```
/*-----Окно поиска-----  
-----*/
```

```
.search {  
    background: var(--color-black);
```

```
padding-top: 1%;

position: relative;
}

.search input {

width: 100%;

height: 44px;

padding-left: 10px;

border: 2px solid var(--color-purple);

border-radius: 5px;

outline: none;

background: white;

}

.search button {

position: absolute;

top: 0;

width: 8%;

height: 100%;

right: -15px;

border: none;

background: var(--color-purple);

border-radius: 0 5px 5px 0;

cursor: pointer;

}

.search button:before {

content: "\f002";

font-family: FontAwesome;

font-size: 16px;

color: white;

}
```

```
.search form {  
  
    position: relative;  
  
    width: 80%;  
  
    margin: auto;  
  
}
```

```
/*-----Окно публикация-----  
-----*/
```

```
.pub {  
  
    display: flex;  
  
    align-items: center;  
  
}
```

```
.pub button{  
  
    position: fixed;  
  
    display: flex;  
  
    align-items: center;  
  
    justify-content: center;  
  
    border: none;  
  
    border-radius: 15px;  
  
    width: 15%;  
  
    height: 7%;  
  
    top: 90%;  
  
    left: 40%;  
  
    background: var(--color-purple);  
  
    cursor: pointer;  
  
}
```

```
.pub a{  
  
    margin: 3%;  
  
    font-size: 23px;
```

```

        color: white;

        content: "\f002";

        font-family: FontAwesome;
    }

    /*-----Луна справа сверху-----*/
    -----*/

    .dark-light {

        color: var(--color-purple);

        margin-left: 88%;

        margin-bottom: 10%;

    }

    .dark-light i {

        font-size: 28px;

        margin: 0;

        cursor: pointer;

    }

```

PYTHON:

```

# Главная страница

@app.get("/", response_class=HTMLResponse)

def read_item(request: Request, db: Session =
Depends(database.get_db), usr_id: Union[str, None] =
Cookie(default=None)):

    if usr_id is not None:

        user = db.query(models.User).filter(models.User.id ==
usr_id).first()

        photos = db.query(models.Photo).all()

        template = "index.html"

        context = {"request": request, 'user': user, 'photos':
photos}

```

```
        return templates.TemplateResponse(template, context)

    return RedirectResponse("/auth") # Перенаправление на страницу
входа
```

```
# Загрузка файлов

@app.post("/uploadfiles")

    async def upload_photos(request: Request, text: str = Form(...),
file: UploadFile = File(...),

                                db: Session = Depends(database.get_db),

                                usr_id: Union[str, None] =
Cookie(default=None)):

        async with aiofiles.open(f"./images/posts/{file.filename}", 'wb')
as out_file:

            content = await file.read()

            await out_file.write(content)


        new_photo = models.Photo(filename=file.filename, content=content,
text = text, user_id=usr_id)

        db.add(new_photo)

        db.commit()

        db.refresh(new_photo)

    return true
```

- Профиль
- Мессенджер
- Лента
- О проекте
- Настройки

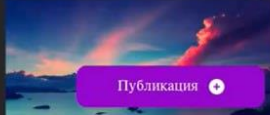
asurasuu



Очень
красивые
цветы

Html weather widget

asurasuu



Публикация



asurasuu



Выберите фото: No file selected.

Подпись:

	Size	Type	Modified
5.png	24.3 kB	Image	2 Apr
6.png	25.7 kB	Image	2 Apr
7.png	24.0 kB	Image	2 Apr
8.png	26.8 kB	Image	2 Apr
9.png	54.0 kB	Image	2 Apr
10.png	25.4 kB	Image	2 Apr
11.png	24.5 kB	Image	2 Apr
12.png	14.1 kB	Image	2 Apr
13.png	23.4 kB	Image	2 Apr
14.png	20.4 kB	Image	2 Apr
15.png	1.8 MB	Image	2 Apr
16.jpg	534.9 kB	Image	2 Apr
17.jpg	536.4 kB	Image	2 Apr
18.jpg	195.3 kB	Image	2 Apr
19.jpg	409.8 kB	Image	2 Apr
20.jpg	382.3 kB	Image	2 Apr
21.jpg	115.4 kB	Image	2 Apr
22.jpg	471.5 kB	Image	2 Apr
23.jpg	1.1 MB	Image	2 Apr
24.jpg	234.2 kB	Image	2 Apr
25.jpg	1.2 MB	Image	2 Apr
26.jpg	156.7 kB	Image	2 Apr
27.jpg	434.4 kB	Image	2 Apr
28.jpg	276.7 kB	Image	2 Apr
29.jpg	360.2 kB	Image	2 Apr
30.jpg	259.5 kB	Image	2 Apr
31.jpg	324.9 kB	Image	2 Apr
32.jpg	260.0 kB	Image	2 Apr
33.jpg	375.8 kB	Image	2 Apr
34.jpg	588.3 kB	Image	2 Apr
35.jpg	536.2 kB	Image	2 Apr

В результате, эта страница отображает ленту новостей, где мы так же можно добавить свою новость и другие пользователи это увидят.

Смена цветовой темы:

JavaScript:

```
// Функция для установки темы

function setTheme(theme) {

    document.body.className = theme;

    localStorage.setItem('theme', theme);

}

// Загрузка сохраненной темы при загрузке страницы

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

    const savedTheme = localStorage.getItem('theme');

    if (savedTheme) {

        setTheme(savedTheme);

    }

});

// Обработчик для переключения темы

document.getElementById('themeToggle').addEventListener('click',
function() {

    const currentTheme = document.body.className;

    const newTheme = currentTheme === 'light-theme' ? 'dark-theme' :
'light-theme';

    setTheme(newTheme);

});
```

CSS:

```
/*-----Тёмная и светлая тема-----*/

body.dark-theme {

    --body-font: "Manrope", sans-serif;

    --color-purple: #9508CD;

    --color-black: #0b1114;
```



```

--color-white: white;

background-color: #0b1114;
}

body.light-theme {
  --body-font: "Manrope", sans-serif;
  --color-purple: #9508CD;
  --color-black: white;
  --color-white: black;
  background-color: white;
}

body{
  --body-font: "Manrope", sans-serif;
  --color-purple: #9508CD;
  --color-black: #0b1114;
  --color-white: white;
  --color-threeCol: #0b1114;
  --color-moon: #9508CD;
  background-color: #0b1114;

  font-family: var(--body-font);
  transition: background-color 0.3s ease;
}

*{
  margin: 0%;
  font-family: "Poppins", sans-serif;
}

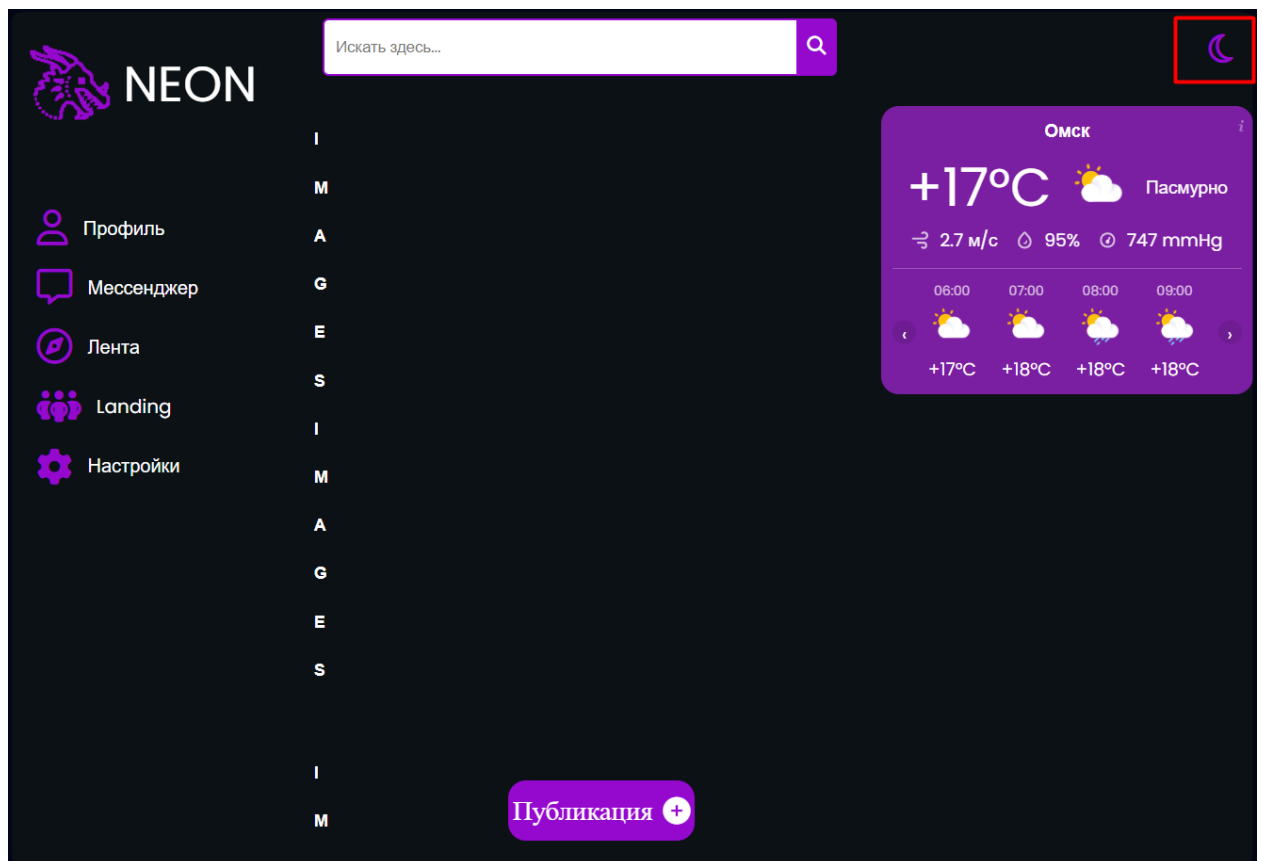
```

```

p{
    margin: 15%;
    font-size: 14px;
    font-weight: bold;
    color: var(--color-white);
    font-family: var(--body-font);
}

a {
    font-size: 14px;
    margin: 15%;
    text-decoration: none;
    color: var(--color-white);
}

```





Встречай новых друзей, делай открытия и оставайся всегда на волне высоких эмоций!

До встречи в следующем приключении, где NEONinst продолжит вдохновлять и удивлять тебя! Всегда будь в центре событий и пульсируй в ритме новостей и общения!

С любовью и энтузиазмом,

Команда NEONinst



ОМСК

2024

