МГТУ им. Н.Э. Баумана

Факультет «Информатика и системы управления»

ДИСЦИПЛИНА:

«Разработка интернет приложений»

Отчёт по лабораторным работам №1-8

и домашнему заданию

|  |
| --- |
| Выполнил:  Группа ИУ5-52Б  Лобанов В.С.  Преподаватель:  Гапанюк Е.Ю. |

2020г

1. **Задача**

Необходимо разработать сайт в формате социальной сети с возможностью создания постов, обзоров, новостей, их изменения, дополнения и настройкой доступа. В него будут включены следующие приложения:

* Система пользователей сайта
* Группы пользователей (роли)
* Пользовательские права
* Личный блог

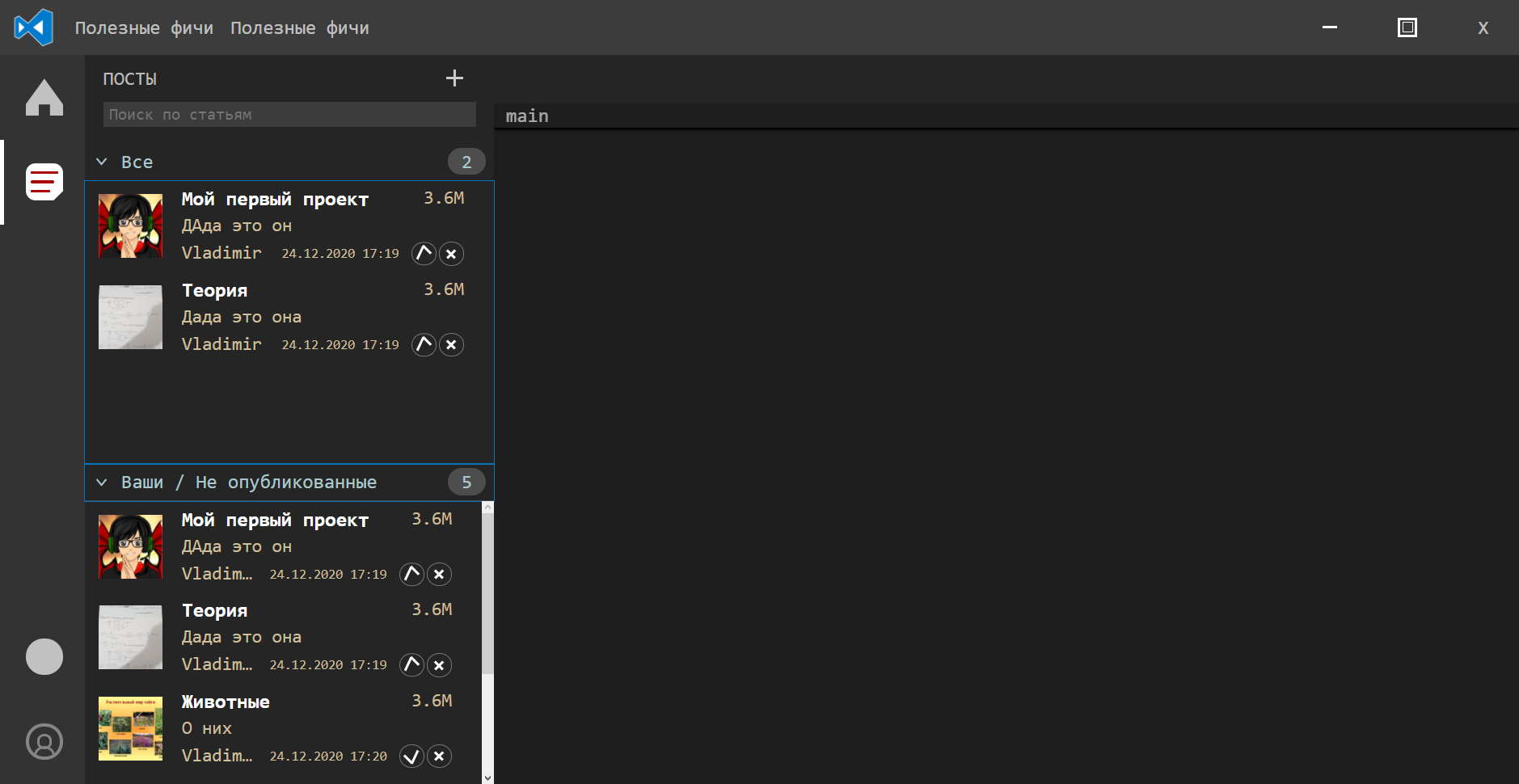
1. **Вёрстка**

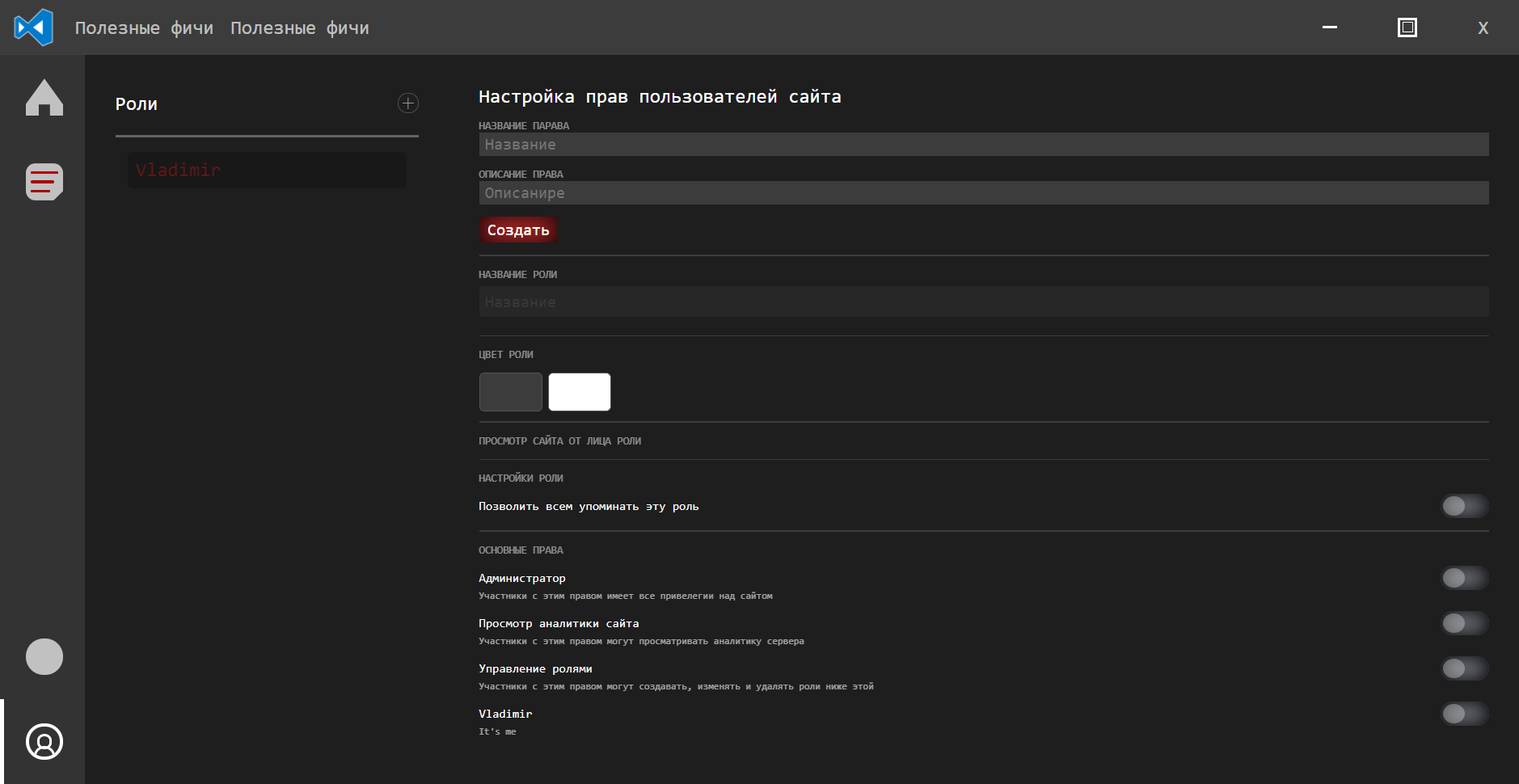
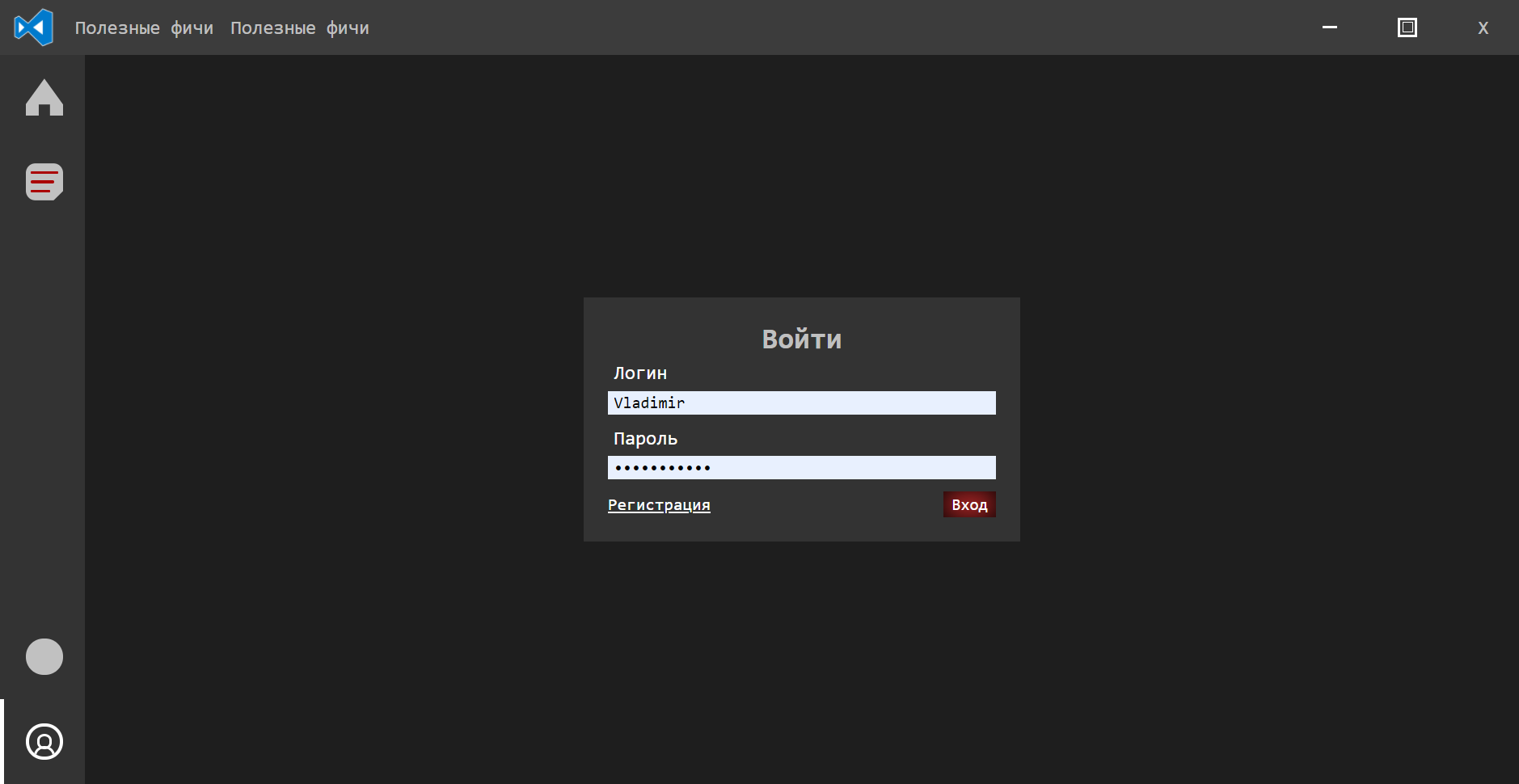
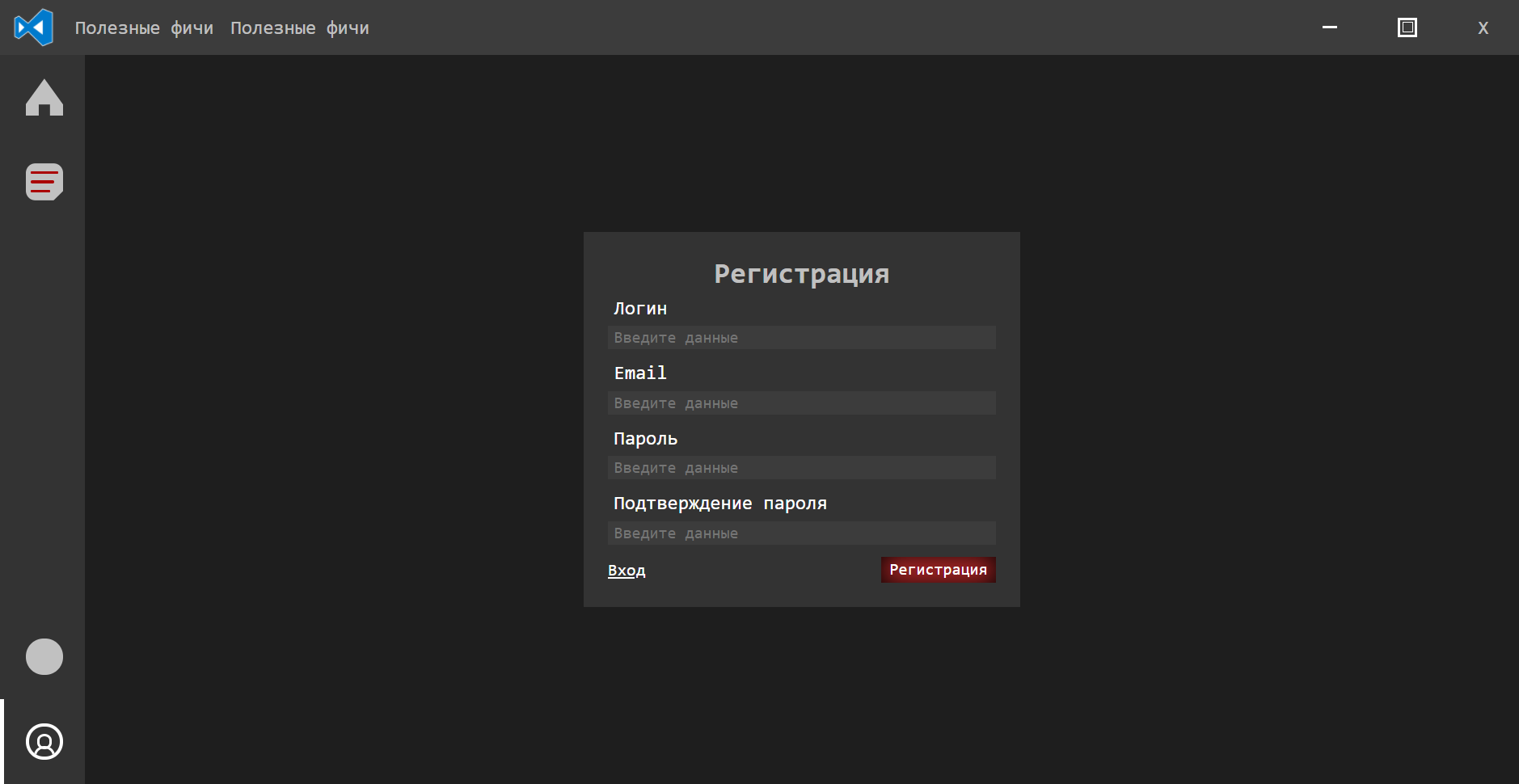
Дизайн сайта разработан на основе таких популярных приложений как Discord, VK, Visual Studio Code и некоторые свои идей. Верстка сайта происходит вручную без использования фреймворков.

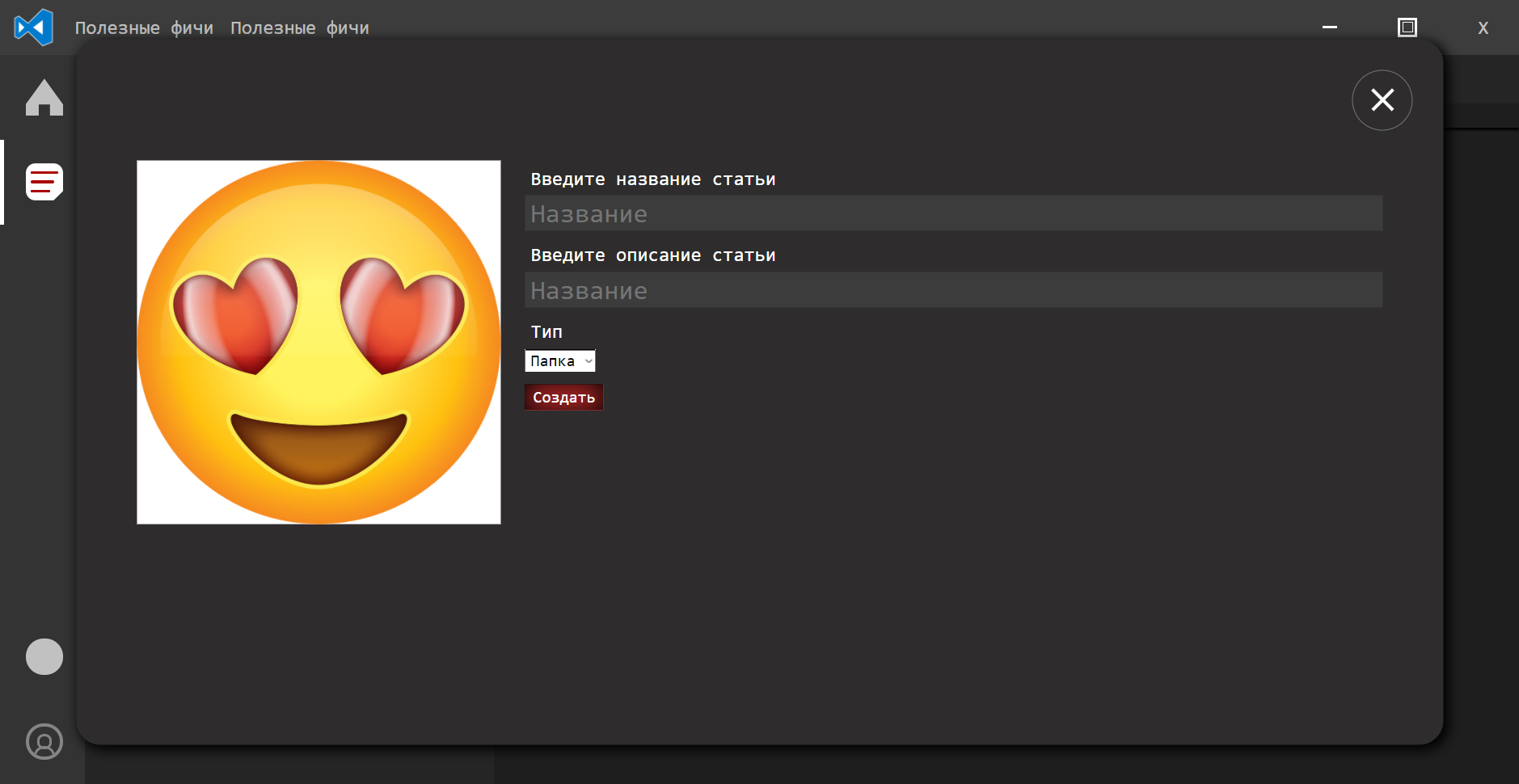
1. Страница пользователя



1. Страница обзоров, новостей, статей



1. Страница настройки ролей   
   
2. Страница вход  
   
3. Страница регистрации  
   
4. Панель добавления новости



1. **Модели**
2. Таблица с текущими обзорами, новостями.

TYPE\_PAPER = (

        ('Папка', 'Папка'),

        ('Файл', 'Файл'),

    )

def upload\_to(instance, filename):

    print(instance)

    return 'images/{username}/{filename}'.format(username=instance.user, filename=filename)

class post(models.Model):

    img = models.ImageField(upload\_to = upload\_to, null=True, blank = True, unique = True)

    title = models.CharField(max\_length = 100)

    intro = models.CharField(max\_length = 300, blank=True)

    text = models.TextField(blank=True)

    creation\_data = models.DateTimeField(auto\_now\_add = True)

    data\_last\_child = models.DateTimeField(auto\_now\_add = True)

    type = models.CharField(max\_length = 30, choices=TYPE\_PAPER)

    user = models.ForeignKey(user, on\_delete=models.CASCADE)

    parent = models.ForeignKey('post', on\_delete = models.CASCADE, null=True, blank = True)

    public = models.BooleanField(default=False)

    def \_\_str\_\_(self):

        return self.title

    def data(self):

        return '{}'.format(self.creation\_data.astimezone(tz).strftime('%d.%m.%Y %H:%M'))

1. Таблица пользователя и его ролей и прав.

def upload\_to(instance, filename):

    print(instance)

    return 'images/{username}/{filename}'.format(username=instance, filename=filename)

class user(models.Model):

    img = models.ImageField(upload\_to = upload\_to, null=True, blank = True, unique = True)

    nickname = models.CharField(max\_length = 50, blank=True, unique = True)

    minitag = models.CharField(max\_length = 50, blank=True, unique = True)

    email = models.EmailField(max\_length = 50, unique = True)

    password = models.CharField(max\_length = 50)

    data = models.DateTimeField(auto\_now\_add = True)

    roles = models.ManyToManyField('role', blank=True, related\_name='roles')

    # collabrators = models.ManyToManyField('collaborator', blank=True, related\_name='collabrators')

    # buys = models.ManyToManyField('buy', blank=True, related\_name='buys')

    def \_\_str\_\_(self):

        return self.nickname

    def get(self):

        arr = [self.nickname, self.email,  self.password, self.time()]

        return arr

    def getAll(self):

        arr = [self.nickname, self.email,  self.password, self.time(), self.roles.all()]

        return arr

    def time(self):

        return self.data.strftime('%d.%m.%Y %H:%M')

    def getRoles(self):

        return self.roles.all()

    def getRolesArray(self):

        arr = set()

        current = self.roles.all()

        for rol in current:

            arr.add(rol.name)

        return arr

    def getFreedoms(self):

        arr = set()

        roles = self.getRoles()

        for i in roles:

            freedoms = i.freedoms.all()

            for free in freedoms:

                arr.add(free.name)

        return arr

class role(models.Model):

    name = models.CharField(max\_length=50)

    settings = models.CharField(max\_length=50, blank=True)

    freedoms = models.ManyToManyField('freedom', blank=True, related\_name='freedoms')

    def \_\_str\_\_(self):

        return self.name

class freedom(models.Model):

    name = models.CharField(max\_length=50, unique = True)

    description = models.CharField(max\_length=200)

    def \_\_str\_\_(self):

        return self.name