WEB DEVELOPMENT PROJECT PROTECTION

Our project is called Hadzen, the link goes to Yandex Zen, our site is designed for reading and creating blogs and posts, you can like, subscribe to channels and watch posts of a separate channel, write comments and read them, there is also a registration and login page.

Хадзен

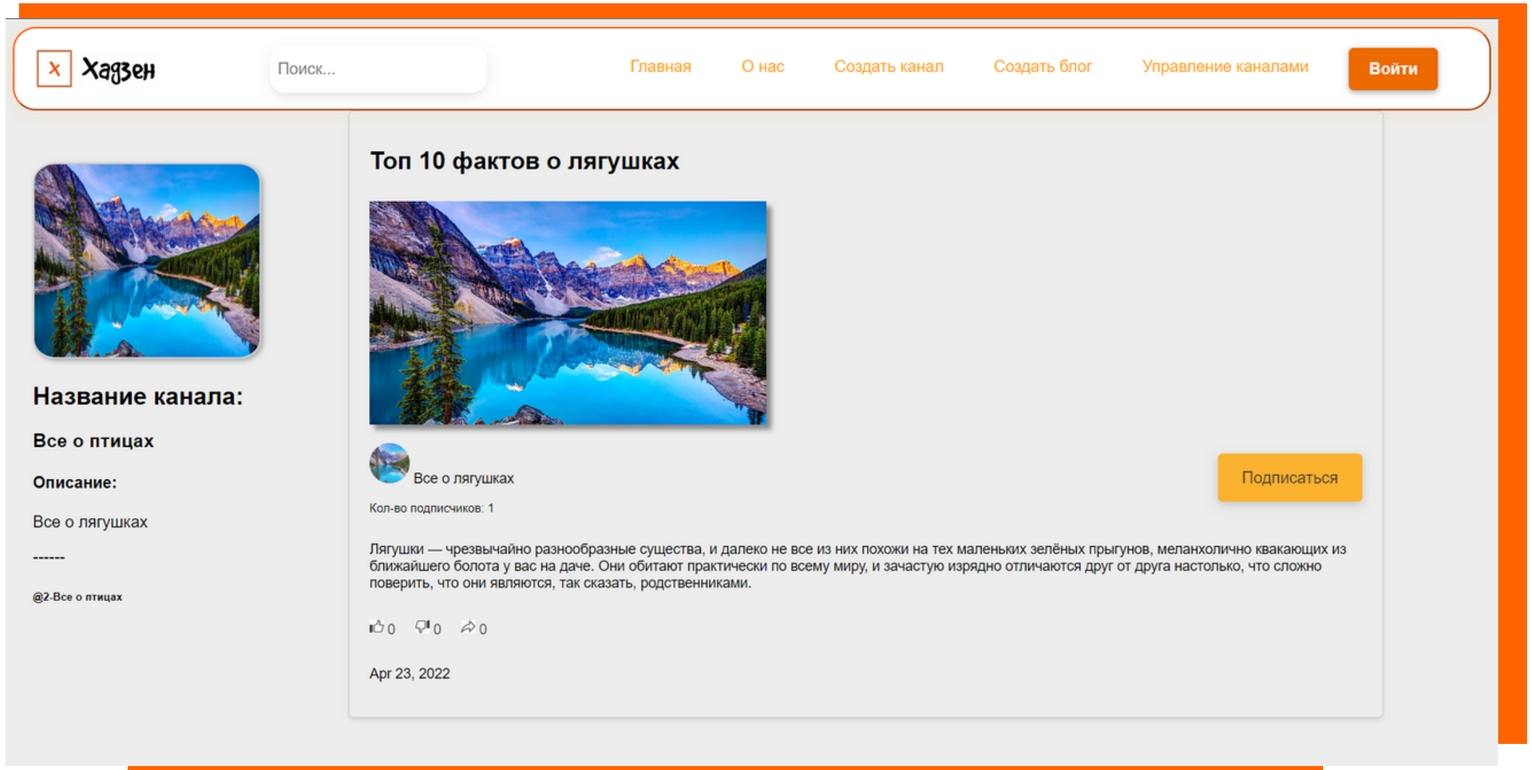
Completed:

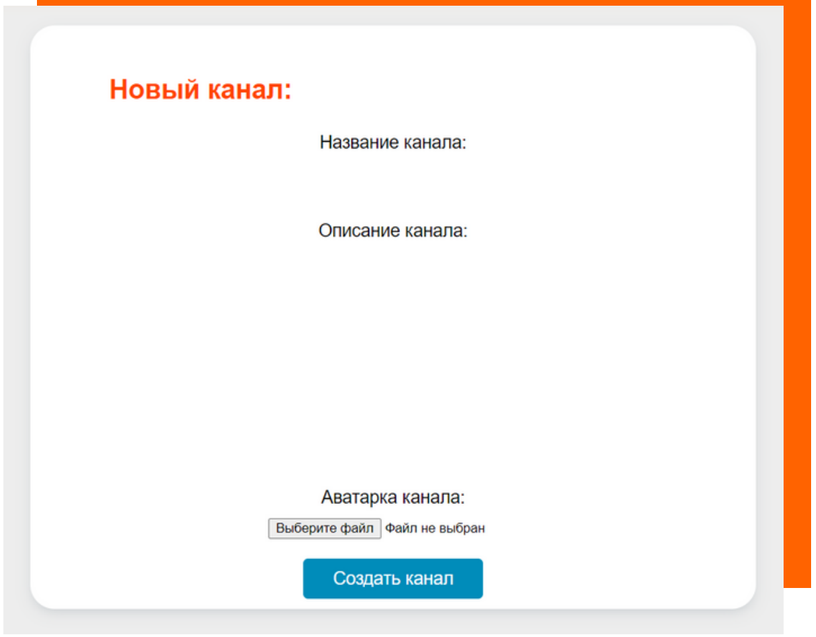
Kusanov Ramses

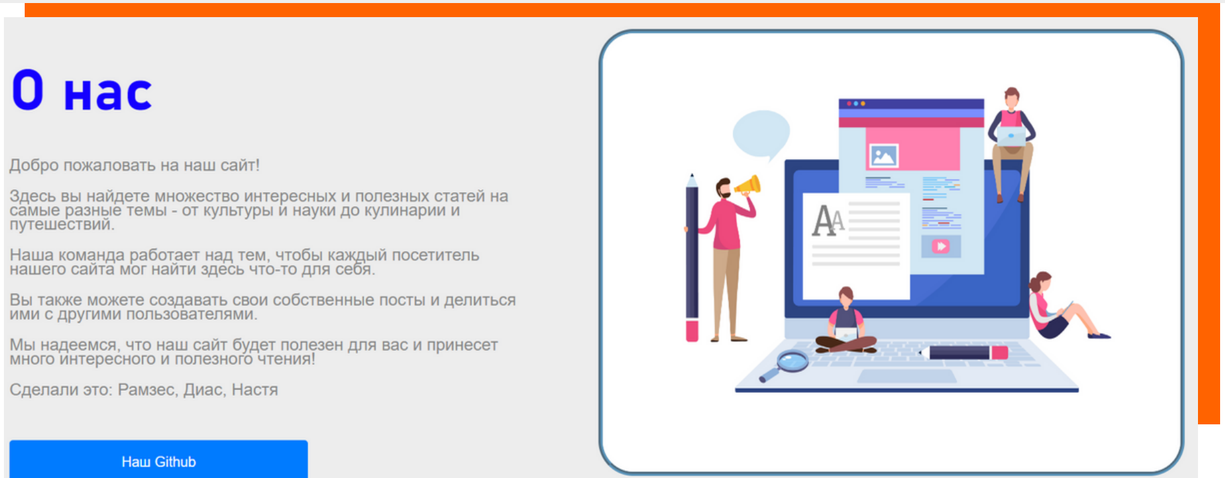
Mombekov Dias

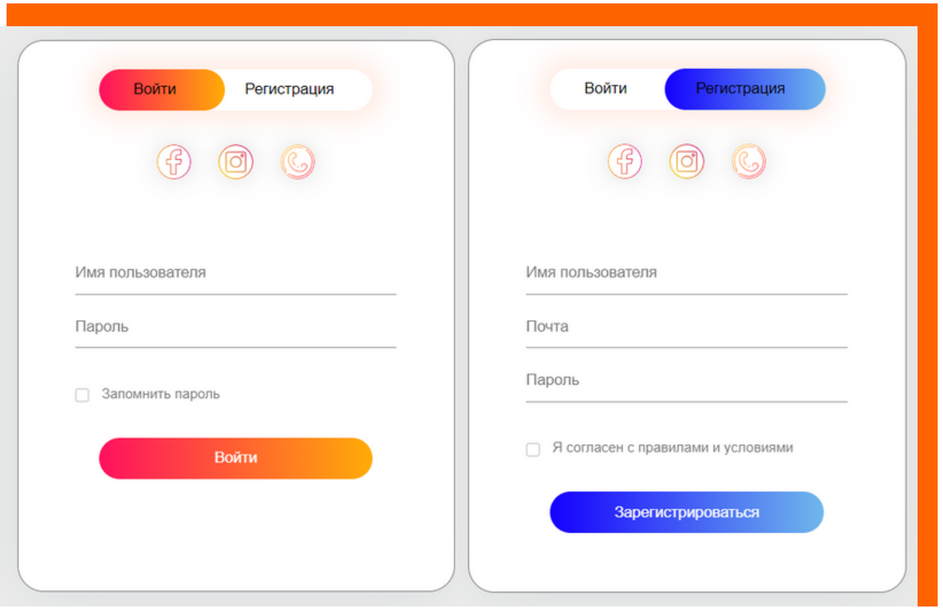
Larionova Anastassiya

Our website is html and css









Front End

REQUIREMENTS

1. All needed interfaces / classes for APIs from back-end
2. All needed services for getting data from API
3. Minimum 4 (onclick) events, which hits API
4. Minimum 4 [(ngModel)]
5. Minimum CSS styling
6. Configured Routing module
7. Usage of directives: ngFor, ngIf
8. JWT based authentication
    a. interceptor
    b. login
    c. logout

IN OUR PROJECT

1. All interfaces / classes for APIs - channel, blog, user, comment, manager, all details
2. All services for getting data from API - read, update, create for channel, blog, comment, manager, all details
3. 1) Create Blog, 2) Create Manager, 3) Show Login, 4) Show Register and more
4. When create blog - 1) Title, 2) Text, 3) Image, 4) Chanel
5. CSS styling - orange and blue, minimalistic
6. Routing module for all links
7. Usage of directives: ngFor, ngIf - in new-post, log-reg, channel-blogs, post-detail, posts, short-channels
8. JWT based authentication
    a. interceptor
    b. login
    c. logout

Back End

REQUIREMENTS

1. Minimum 4 models
2. Minimum 1 model Manager (optional)
3. Minimum 2 relations between models (ForeignKey)
4. Serializer
    a. at least 2 from serializer.Serializer
    b. at least 2 serializer from serializer.ModelSerializer
5. Views
    a. at least 2 FBV drf view
    b. at least 2 CBV APIView
6. Token based authentication
    - login
    - logout
7. At least for 1 model CRUD operations
    a) create objects with authenticated user (Ex: Tasks for current authenticated user)
8. Postman requests with all implemented methods

IN OUR PROJECT

1. Minimum 4 models - channel, blog, user, comment, all details
2. Minimum 1 model Manager - can create all
3. Relations between models (ForeignKey) in comments, manager, subs, blog and channel
4. Serializer
    a. from serializer.Serializer - blog_ldc, subs_count
    b. from serializer.ModelSerializer - user, channel, blog_all, comm, subs
5. Views
    a. at least 2 FBV drf view - all lists
    b. at least 2 CBV APIView - all lists
6. Token based authentication
    - login
    - logout
7. At least for 1 model CRUD operations
    a) create objects with authenticated user
8. Postman requests with all implemented methods