

GDPR Document

GDPR Document for U2.

Created by:

Petar Bakalov – 4634705

Contents

Versioning table	1
Introduction	2
Data Security	2
Right to Erasure (to be Forgotten)	2

Versioning table

Version	Date	Description
1.0	20/06/2024	Initialization of document

Introduction

GDPR (General Data Protection Regulation) is a comprehensive data regulation law that regulates how personal data of individuals within the European Union (EU) is collected, processed, stored, and transferred.

As commitment to data privacy and protection is crucial, while developing an enterprise software solution, this document aims to provide transparency, while explaining some key GDPR rules, that are relevant for “U2”.

Data Security

For handling sensitive user data, such as email and password, “U2” uses Auth0 as a third-party service. Auth0 implements robust security measures, including encryption, access controls, and regular security assessments, to protect personal data. Moreover, it was decided that, for extra security measures, only that information will be handled using Auth0. For example, information like uploaded videos, posted comments, etc., will be handled separately, using our own database.

Auth0 also informs the user what information will be gathered on sign up, and how it will be used. This also covers the “There is a right for the user to see what data the company has collected” rule.

Right to Erasure (to be Forgotten)

Other important GDPR rule for the context of “U2” is the “Right to be forgotten”. Every user has the right to delete their account and all their information on the platform, including videos and posts.

While researching other available products, such as YouTube, I discovered that they actually have some rights on every video posted on their platform.

Therefore, you can never actually delete a video from their records, only make it private, for example.

However, I decided to fully comply with this GDPR rule, and using a message broker, all videos for a user are deleted, when their account is deleted. This also proves that a user can truly be forgotten, erasing all their information on the platform. Below I have attached some screenshots of the implementation of the message broker and proof that it works as intended:

```
2 references | Petar Bakalov, 4 days ago | 1 author, 5 changes
public void Consume()
{
    Console.WriteLine(" [*] Waiting for messages.");

    var consumer = new EventingBasicConsumer(_channel);
    consumer.Received += (model, ea) =>
    {
        var body = ea.Body.ToArray();
        var message = Encoding.UTF8.GetString(body);
        var deserialized = Newtonsoft.Json.JsonConvert.DeserializeObject<User>(message);
        Console.WriteLine($" [x] Received {deserialized?.Id}");
        _videoService.DeleteVideosForUser(deserialized?.Id);
        Console.WriteLine($" [x] Deleted videos for user: {deserialized?.Username}");
    };
    _channel.BasicConsume(queue: "delete_user",
        autoAck: true,
        consumer: consumer);
}
```

Fig. 1 "Implementation of RabbitMQ message broker"

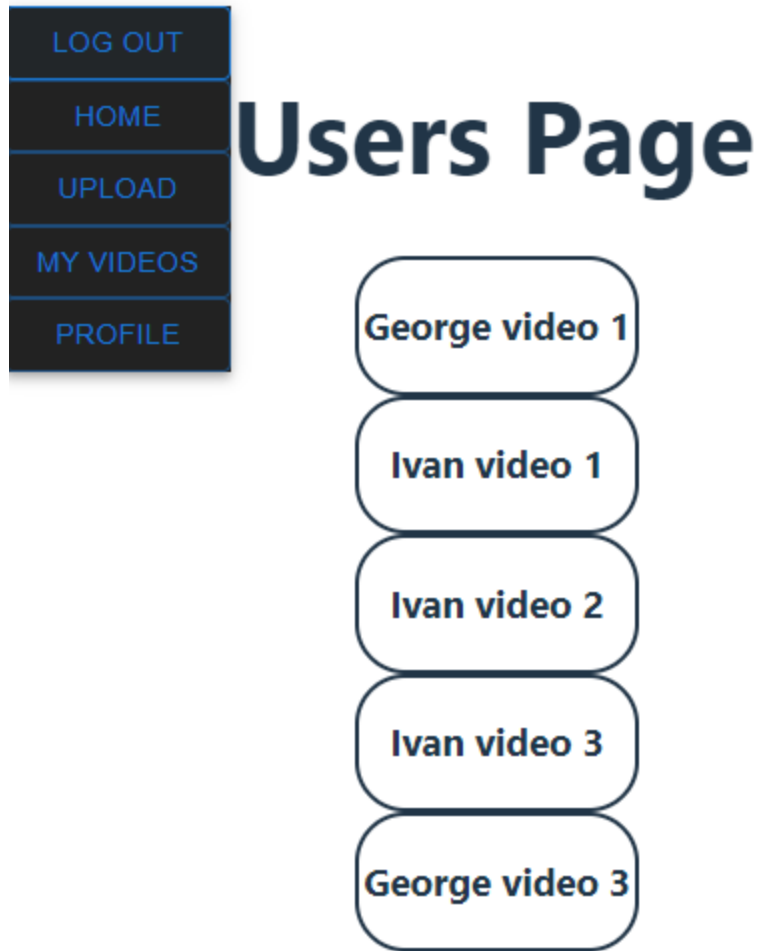


Fig. 2 "Initial list of all videos, viewed from the profile of user 'George'"

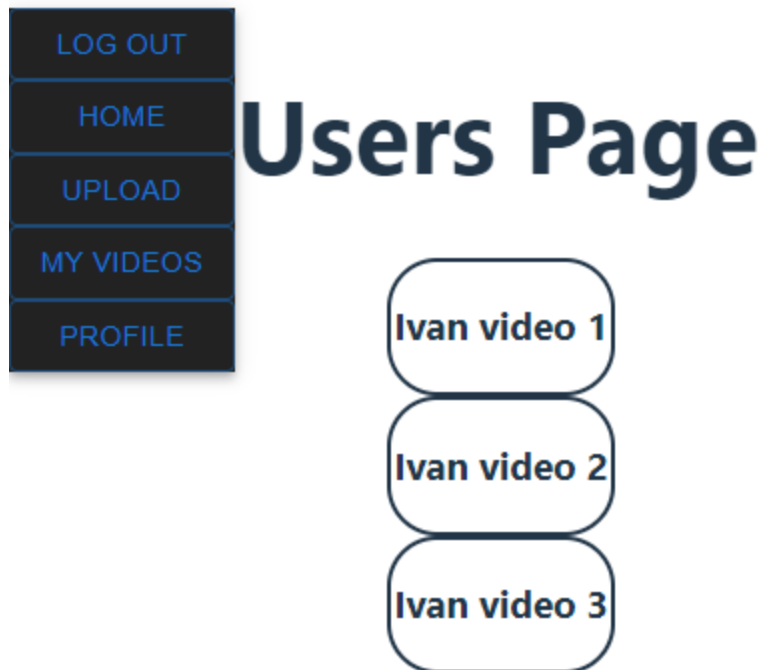


Fig. 3 "List of all videos, after George deleted their account"

```
C:\FHICT\S6\Individual\U2\u2-client\src\test>kubectl logs videoservice-7d79579878-9dgsd
[*] Waiting for messages.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:8080
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app
Sending message!!!!
Sending message!!!!
Sending message!!!!
[x] Received 11
```

Fig. 4 "Logs from the service, that deleted the videos of the user"