**Short Report**

**System Description**

This project is a Blazor WebAssembly application designed to allow users to search for YouTube videos, detailed information about individual videos and navigate paginated search results. The system integrates with the YouTube Data v3 API to dynamically fetch video data and is hosted on Azure.

**Key Features**

* A responsive search bar for querying YouTube videos.
* Paginated results with video thumbnails, titles, and descriptions.
* A dedicated video details page showing additional information like video description, channel name, and published date.
* Input validation and error handling to improve user experience.
* Testing using Playwright.

**Screenshots**

**Search Results Page**

**Obraz zawierający tekst, Ludzka twarz, zrzut ekranu, człowiek

Opis wygenerowany automatycznie**

**Video Details Page**

**Obraz zawierający tekst, zrzut ekranu, Ludzka twarz, osoba

Opis wygenerowany automatycznie**

**Deployed Application and Repository URL**

You can access the deployed application and github repository here:  
**Deployed App URL:** <https://ca3hosting.z16.web.core.windows.net>

**Repository URL:** <https://github.com/DawidRatajczyk02/CA3_Blazor_WASM>

**Description of Tests Performed**

Automated testing was performed using Playwright, targeting key functionalities of the application to ensure reliability. Tests include:

1. **Search Functionality Test**
   * Verifies that entering a query in the search bar displays six results per page.
   * Validates that the search bar cannot be empty.
2. **Pagination Test**
   * Tests the "Next" and "Previous" buttons to ensure paginated navigation works as expected.
   * Confirms that clicking "Next" shows a new set of results and that "Previous" navigates back correctly.
3. **Video Details Test**
   * Ensures the video details page loads with correct video information.
   * Verifies that the "Back to Search" button redirects to the search results page.

**Code Quality and Metrics**

**Code Structure**

* **Separation:**
  + Razor pages handle the UI, while C# services fetch YouTube API data.
* **Validation and Error Handling:**
  + Includes input validation for search queries and displays user-friendly error messages.
  + Errors in fetching data are caught and displayed to users.

**Best Practices**

* The code is modular, with clearly defined methods for search and API interactions.
* Includes comments explaining the purpose of key functions.