

# Summify

## User Manual

Project Title:	
Summify	
Supervisor: Dr Jennifer Foster	

Group Members	
Benjamin Olojo	19500599
Przemyslaw Majda	20505049
Submission Date	24/02/2023

## Table of Contents

1. Abstract	3
2. System Overview	3
3. Dependencies	3
4. Installation	3
5. User Functionality	4
5.1 The Home Page	4
5.2 Entering YouTube URL	4
5.3 Using the Summaries and Related Links	5
5.3.1 Related Links	6
5.3.2 Summaries	7
5.4 Development Blog	7

## 1. Abstract

This document aims to provide a detailed guide on how to operate the Summify web application. It will highlight each aspect of the application both graphically and through text to produce an in-depth walkthrough of its functionality.

## 2. System Overview

The Summify web application is a single-page web application (SPA) which allows any user to generate a summarised version of the transcript of YouTube videos that have a transcript publicly available. The application also displays Wikipedia links to pages related to the video which the user can use for further research. The application has a minimalistic interface, giving the user only simple instructions and a text input field for the YouTube video URL. The input field prompts the user for a YouTube link, and the page displays various error messages if the user does not input a valid link. The application also has a link to view a development blog outlining the development process.

## 3. Dependencies

Assuming that the user has access to a working electronic device capable of internet connection, the main requirements for the SPA are as follows:

- A viable internet connection
- A modern web browser
- Access to YouTube/YouTube video URL

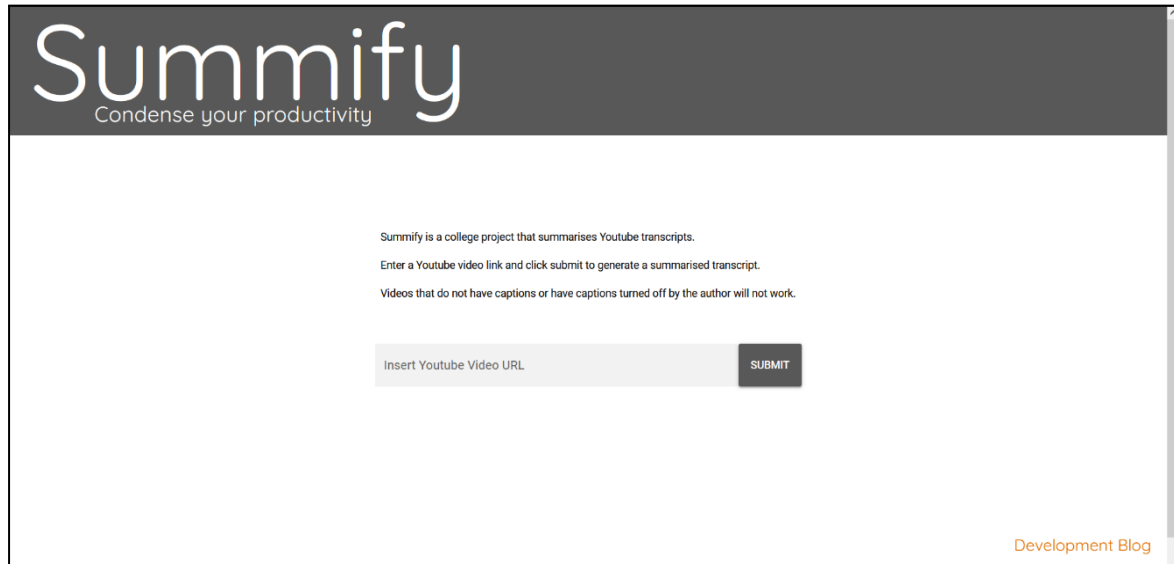
## 4. Installation

There are no installations required for a user to access Summify. The user needs only to visit the domain on a web browser of their choice.

## 5. User Functionality

### 5.1 The Home Page

When the user visits the homepage, they will see a screen with simple instructions and a text input field. The input field is for a video URL, where the user can enter a link to a YouTube video. There is also a link at the bottom right of the page which leads to the development blog page, where the user can read about the overall development process.



### 5.2 Entering YouTube URL

If the user enters a URL of a video that does not have a transcript available or has a transcript too large for the application to process, a message will appear explaining why the application was not able to complete the request.

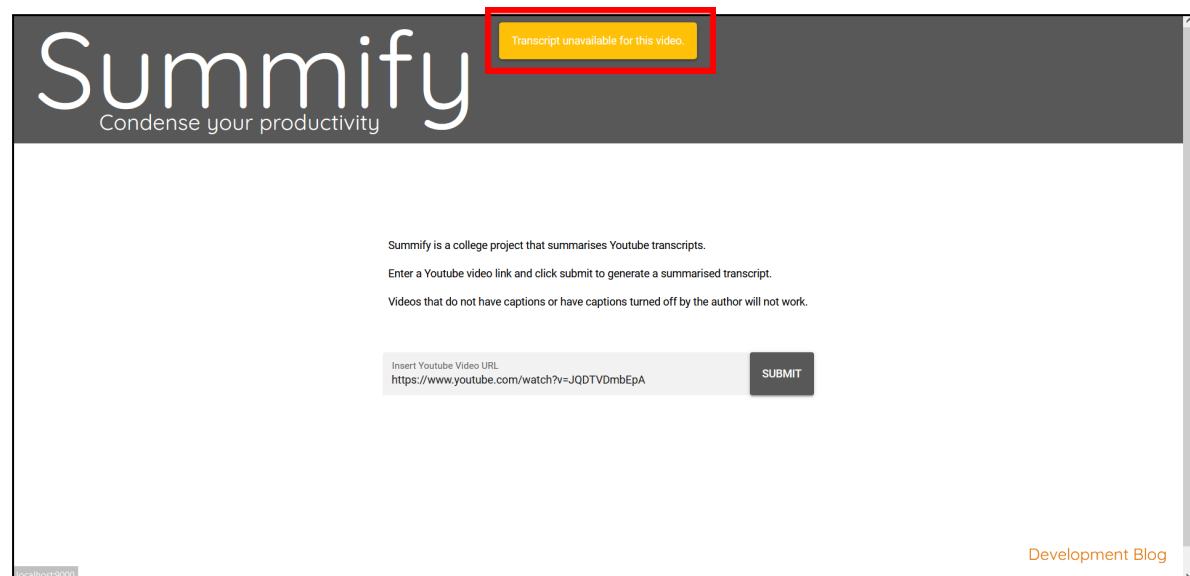


Figure 2 Transcript unavailable error message

If the user enters a valid URL, the application will enter a loading state to let the user know the request is being processed. A loading spinner will appear in the centre of the screen as well as a loading bar at the bottom to indicate the progress of the summary generation.

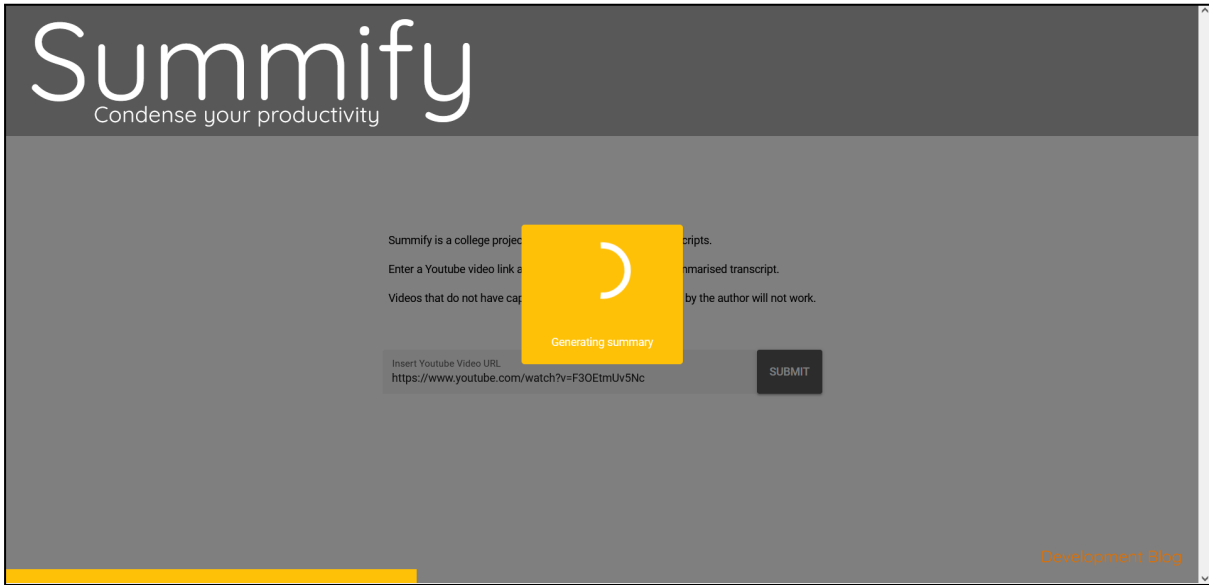


Figure 3 Generating summary loading state

5.3 Using the Summaries and Related Links

Once the request completes, an overall summary will be displayed along with segmented summaries which go into more detail about each 5-minute segment of the video. The video will also be embedded on the left-hand side of the screen for the user to not need to switch to a different tab and be able to watch/listen while they read the summaries. Under the embedded video, links to related topics will also be displayed. These are Wikipedia links to topics related to the video, and when clicked they will open a new tab.

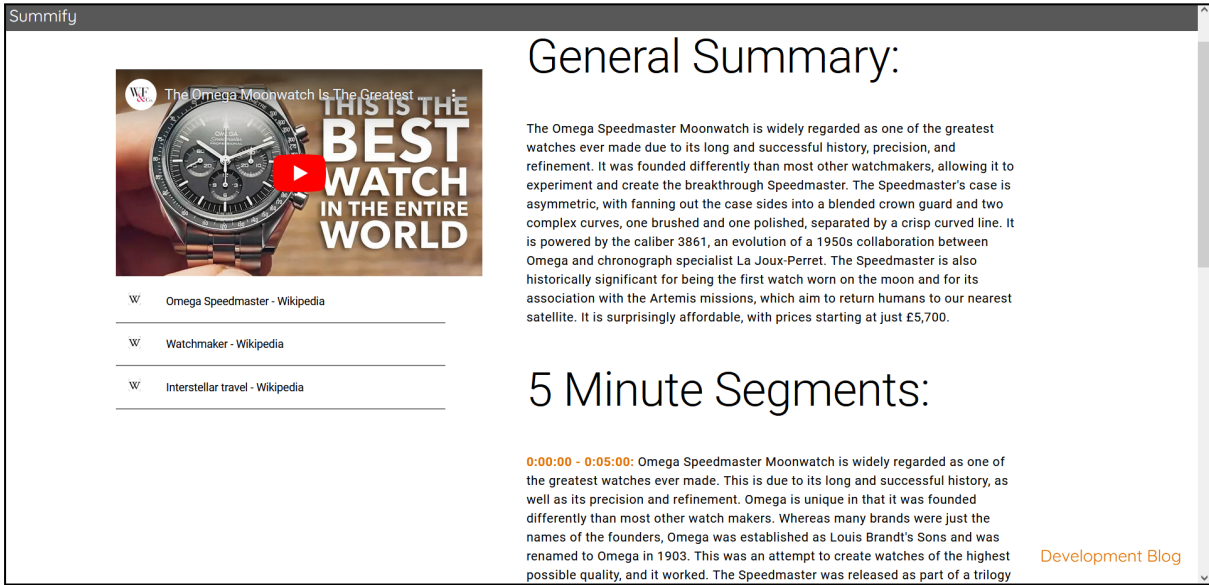


Figure 4 Summaries and links displayed

5.3.1 Related Links

The user can click on the links displayed under the video embed which will open a new tab. The original Summify tab will stay open.




Figure 5 Clicking link

### 5.3.2 Summaries

The user will have a summary for each 5 minute segment of the video which they can select and copy if they want to paste the text into an editor or for any other use they might have

Summify



W

Omega Speedmaster - Wikipedia

W

Watchmaker - Wikipedia

W

Interstellar travel - Wikipedia

founded differently than most other watchmakers, allowing it to experiment and create the breakthrough Speedmaster that is still highly regarded today.

**0:05:00 - 0:10:00:** The Omega Speedmaster is an iconic watch that is known for its bold design and high quality. It was designed to be operable at the speeds its tachymeter was capable of recording, and Omega refused to compromise on quality. The Speedmaster's case is asymmetric, with fanning out the case sides into a blended crown guard and two complex curves, one brushed and one polished, separated by a crisp curved line. Inside the Speedmaster is the caliber 3861, which is an evolution of a 1950s collaboration between Omega and chronograph specialist La Joux-Perret. It is one of the few hand-wound chronograph [calibers still produced in Switzerland](#). The Speedmaster is historically significant for being the first watch worn on the moon, and is now associated with the Artemis missions, which aim to return humans to our nearest satellite. The mission to the moon and the Space Race began in 1961 with Project Mercury, just four years after the Speedmaster first debuted. In just eight short years, Project Mercury, Gemini, and Apollo went from an initial experiment to see if humans could survive in space to actual space travel. The Speedmaster is a testament to the incredible progress that has been made in space exploration.

**0:10:00 - 0:12:55:** The Omega Speedmaster is a timepiece that has been famously worn by Neil Armstrong on the Apollo 11 mission to the moon in 1969. This was a significant event in human history as it marked the first time that humans had ever set foot on the moon. The Speedmaster is also known as the "Moonwatch" due to its association with the first lunar mission, and it is truly a remarkable piece of engineering. It is also surprisingly affordable, with prices starting at just £5,700. This makes it one of the best values in the watch market.

Development Blog

Figure 6 Highlighting summary text

### 5.4 Development Blog

There is a link at the bottom right of the page for the user to click which will bring them to the development blog where they can read about the development of the app.

Summify

28/12/22

Having only completed one full-stack project, our knowledge of leading platforms and frameworks was quite low. We were originally going to go with Django for the backend and Node.js for the frontend, but after some research decided to go with Flask for the backend, and with Quasar for the frontend.

Our reasoning was that our web app will be a single-page-application (SPA) without registered users, so we would not need all of the heavy lifting features that Django provides (admin panel, database, authentication etc.).

Flask is lightweight and flexible and provides simple HTTP request handling which was perfect for us. Quasar is a frontend Javascript framework built on Vue.js, and includes many built-in components and functionality (loading bars, spinners etc.) which we thought would speed up and optimise our development process. In our research we had learned about Vue's virtual DOM and dynamic rendering of HTML elements, which would be good for a SPA where we do not have any routes and all of the work is done on the one page. From the brainstorming part of the project we had the vision for our page to not have to reload when a request is processed, and Quasar would let us do that.

02/01/23

This week we focused on outlining the different prototypes we would have to build during the initial stages of the implementation phase. To do this, we revisited our functional specification and highlighted the several components that the application would entail as well as the functionality.

We determined that prototypes for backend components would be created within Jupyter notebook and used as the building blocks of the application. These prototypes included obtaining the video transcript through the YouTube Transcript API, generating the transcript

Development Blog

Figure 7 Summify development blog page