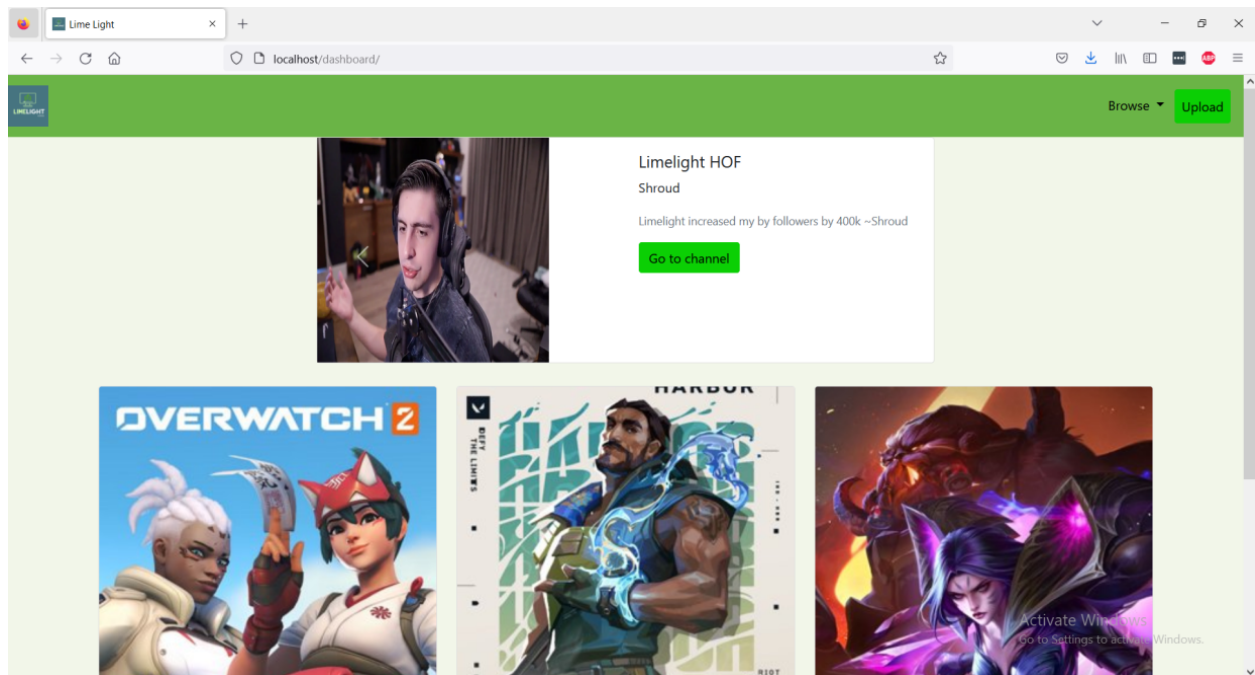


Project Streamer Discoverability



This project aims at helping low view twitch streamers promote their streamers on our website and grow their viewers naturally and help build their community.

1.1 Product Purpose

The benefit of this product will be that users who are looking for small streamers to watch won't have to scroll through multiple streams in order to get to those streamers. Another benefit is that small streamers will be able to grow their Twitch channel and increase the discoverability of their Twitch stream. One objective of this product is to create a convenient and easy method for a person to find new small Twitch streamers to watch. Another objective is to create a simple and easy way for a small Twitch streamer to increase the discoverability of their stream and grow their Twitch channel. The main goal of this product is to have a functioning website that displays multiple stream categories with each category having subcategories that

will lead to the URLs of small Twitch streamers. The intent is to promote the discoverability of small streamers by making it easier for people to find their Twitch streams without hassle.

1.2 Product Scope

The current method of finding small content creators on Twitch is by choosing a category and scrolling until you find a stream that seems appealing. This initial scrolling until you find content creators with less than 100 viewers takes a very inconvenient amount of time. Every time you revisit the Twitch website, you must redo this initial scrolling to find small streamers. This project proposal will help users of twitch both new and old find really small content creators much faster than this current method. This will help small content creators as well potentially grow their audience faster than the traditional method of streaming and hoping viewers appear through word of mouth or with viewers going through the current stream finding method.

1.3 Intended Audience

The target audience for our products is mainly 2 groups which consist of small streamers and people who want to watch smaller streamers. Our product achieves this by promoting streams with low views to the top of the page helping to promote small streamers. By doing this it also helps viewers find new and up incoming streamers to support and watch.

1.4 Intended Use

The website will be similar to Twitch; however, there will be a huge difference in that it will be divided into different categories, and within those categories will be Twitch URLs that link directly to videos with a low number of viewers. In order to have their URLs appear on the website, streamers will upload their Twitch URLs to the website and be placed in a queue in which they will wait for their turn to have their URLs posted on the website. Therefore, when viewers visit our website, by clicking on their favorite categories, they can choose any URLs to support streamers.

1.5 General Description

A platform that promotes and advertises for small Twitch streamers (2-50 subscribers).

The website will have a main page that gathers the users preferences in content, after gathering this information, the user will be lead to another page that contains links to the streamers twitch account. At the bare minimum this website will be a fully functional website with the functions of filtering content creators by category, and promoting their accounts on a display page. If we accomplish this, some add-on features we would like to include if possible are:

- a featured/hot list that displays trending streamers gaining the most traffic.
- a 'Like/dislike' button that can be applied to better provide desired content.
- a profile feature where a user can create an account that is linked to all their data.

The features listed here are liable to change as we progress through the project.

The add-on features are all supplementary and will only be attempted once we have a functional website.

2.1 Product Perspective

Our product is built to help enhance an existing product in a streaming service called *Twitch*. The overall goal of our product is to help viewers find streamers to watch that have a smaller following without having to dig through Twitch's UI scrolling through multiple 0 viewer streams. Our system works by gathering data from Twitch's website and uploading links to twitch streams that meet the users criteria.

2.2 Product Functions

Our product has a few major functions it must complete to be considered a working product. The first main task our product must complete is gather data from Twitch on the streamer which includes if they are live, what they are streaming, and their streamer name. The

other main task our website must complete is to show a document with links to these random streams that the user may want to watch and give them the information gathered from twitch so they can make an informed decision on which stream to choose to watch. Our product also needs to save all the information gathered into a database we and our users can access. Lastly our product needs to be able to accept link submissions from streamers so we can add their channel link in our database so we can promote it to our users.

2.3 User Classes and Characteristics

Our user classes are simply broken down into two groups which are viewers and streamers. The first class who are viewers will be the main user of our product. This is because viewers would be using our site to help find new streamers in the categories they specify until they find a stream they want to watch. The second class would be the streamers who would access our product to submit their twitch channel link we would use to get their stream data that tells us when their live streaming would allow us to add their link to our database that would then push to those looking for a stream to view.

2.4 Operating Environment

The main environment our product would be used in is on Desktop web browsers. Our plan is to start off our product to work on Desktop web browsers and expand to other types of web browser like tablet and phone browsers.

2.5 Design and Implementation Constraints

Some of our product's design and implementation constraints come from being dependent on gathering data from a third party which in our case is *Twitch*. Our product scrapes information on the streamer which they could stop us from accessing at any time. Another limitation on our

end is a memory one since we are storing the streamers twitch link and other information in a database it would take up alot of memory on any system.

2.6 User Documentation

Some of the documentation that our product will provide users would be giving a brief explanation telling those who would want to add their stream link to our data where and how to do so. We would also have a quick tutorial for users on how to navigate and use our product very briefly as it will be very straightforward.

2.7 Assumptions and Dependencies

Some of our products assumptions and dependencies include but are not limited to *Twitch's* website to allow us to gather specific data from its website to use in our product. Some of our dependencies include using cloud services to save our database, accessing twitch data, and having streamers upload their stream links to our product.

3.1 User Interfaces

The user interface our website uses for the project is a very simple and straightforward design. Our UI design is going to look very similar to the service we are trying to enhance which is twitch. Our home screen will only have a couple of buttons that are the 3 main streaming categories streamers use for our users to select from. Also on our front page is an upload link button that pops up when clicked to allow steamers to submit their twitch link to our database. Once the user selects a category the user will be sent to the next page which will link you to our database of live twitch streams for you to select from.

3.2 Hardware Interfaces

The hardware interfaces used in our project would be the different supported devices our product services. In our case it would be devices that can access the internet through a web browser which would be Phones, PC's, Laptops, Tablets any device that can access a web browser.

3.3 Software Interfaces

Some of the software components our product interacts with are numerous, one of which includes a database. Our software interacts with a database that stores the twitch links and some information on the steamer for when the user is pushed to the link can make an informed decision on whether they want to watch. The database the users would interact with would be refreshed manually by the user. The software will also have a general browser that holds our library of stream links stored in our database.

3.4 Communication Interfaces

Some of the communication interfaces used in our product include having different displays with information on our main page of our website. The messaging standards used will be WHATWG HTML5. The other communication method our product would use would be the information we communicate to our user from the database of twitch streams we host which would be through mongodb using NodeJs.

4.1 Stream link drop off

One of our high priority features to have implemented is the ability for streamers to submit their twitch stream link to be saved in our database to be promoted on our website whenever the streamer goes live and our system reads that they are live streaming.

4.2 Database

The other high priority feature our website would have is a database that stores steam links and basic steamer info that would display in a document to our users when they use the service to help them make an informed decision on what new stream to watch.

4.3 Web Scraper

The last high priority feature our website would use is a web scraper to gather information on the twitch streamer which include whether their currently live streaming, the view count, the streamer's name, and the tags the stream is using . This helps us present the streamers to our user to better represent them and allows users to make an educated choice to either view or not view the stream.

5.1 Performance Requirements

The performance requirements of our product are that our database of information regularly and timely updates its information, allows for more information to be added to it at a moment's notice and properly communicates with our website. The other performance requirement is to have our website be only and running at all times for users to access.

5.2 Safety Requirements

There aren't any real safety concerns for our users from our product as we ask for no private or personal information therefore there are not any safety precautions needed to be made on our end other than making our domain https.

5.3 Security Requirements

For our product the only security requirement we have is a captcha prompt for those who wish to upload their twitch link to our database as to slow down or stop someone from automating the process and spamming our database with 1 specific link.

5.4 Software Quality Attributes

Some of the quality attributes our product strives for our users is availability and simplicity. We want our product to be available any time, any where, and on any device with a web browser. The other attribute our product strives for is simplicity for the users to navigate our product.

5.5 Business Rules

We currently do not have any business rules we abide by but if some were to pop up they would be added as needed.