

To create a video Q&A website in order to help potential clients find a professional they vibe with

Richard Evans-Lacey

MSc Computer Science project report

Department of Computer Science and Information Systems, Birkbeck College, University of London

2022

This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I have read and understood the sections on plagiarism in the Programme Handbook and the College web site. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.

The report may be freely copied and distributed provided the source is explicitly acknowledged.

Table of contents

TABLE OF CONTENTS.....	2
ABSTRACT	3
INTRODUCTION.....	4
SYSTEM DESIGN	5
CHANNELS	5
USERS.....	5
CHOICE OF STACK	5
HIGH LEVEL STRUCTURE	6
PAGE STRUCTURE.....	7
RELATIONAL DATABASE MODEL	7
FILE STRUCTURE	8
WORKSPACE SETUP	9
PAGE DESIGN	11
NAVIGATION	11
WELCOME	12
BROWSE VIDEOS.....	16
BROWSE PROFILES.....	18
ASK A QUESTION	19
USER SIGN-UP.....	20
FAVOURITES.....	22
PROVIDER CONTACT.....	22
PROVIDER SIGN-UP	24
RECORD VIDEOS INTRO	26
RECORD VIDEOS	26
MANAGE VIDEOS.....	30
PAGE IMPLEMENTATION.....	31
NAVIGATION	31
WELCOME	32
BROWSE VIDEOS.....	34
BROWSE PROFILES.....	35
ASK A QUESTION	35
USER SIGN-UP.....	35
FAVOURITES.....	37
PROVIDER CONTACT.....	37
PROVIDER SIGN-UP	37
RECORD VIDEOS INTRO	40
RECORD VIDEOS	40
MANAGE VIDEOS.....	43
TESTING	44
UNIT TESTING.....	44
INTEGRATION & USER TESTING	44
REFLECTIONS	48
BIBLIOGRAPHY	49
APPENDICES	53
APPENDIX 1: USER ACCEPTANCE TESTING MEETING NOTES	53

Abstract

The objective of this project is to create a website that helps potential clients find a provider of professional services who they vibe with. It is based on the idea that video content helps potential clients make better 'like' decisions by conveying non-verbal information about the provider.

To explore this idea, I created a website based around short video answers to questions on the theme of mental health. These answers are intended to help inform the user and also to help them get to know providers who they could work with personally.

Users browse through a list of video answers to questions that interest them, like the answers and providers they are drawn to, check out their profiles, and contact them through the site.

The video content is created by the providers on the site. When they are ready to record, the site asks the provider a randomly chosen question, gives them 3 seconds to consider their answer, then starts capturing video using their computer's built-in camera. After a maximum of 60 seconds the recording stops and it is available for them to review. If they are satisfied with their attempt they can choose to add it to their library, the site uploads it to storage in the cloud, and it is available for potential clients to watch.

Supervisor: Vladislav Ryzhikov

Introduction

The objective of this project is to create a website that helps potential clients find a provider of professional services who they vibe with. The project was fully introduced in the project proposal submitted earlier this year (Evans-Lacey, 2022). That document explained the challenges involved in finding a professional services provider who a user feels comfortable with and confident in. It proposed that a website utilising spontaneously recorded video content would help facilitate this decision-making process.

The website created in this project is appropriate for multiple channels but I have initially focussed on mental health as this is an area where finding a good match is particularly important to the outcome of the interaction. I have experience in this context as I have chosen a therapist using existing web-based listing sites. I am a psychotherapist myself so I am familiar with the issue from both sides.

The objective of this project is to create a website that:

- Displays video answers to frequently asked questions,
- Guides users and providers through a simple sign-up process,
- Enables users to save their favourite providers,
- Enables users to contact their favourite providers,
- Enables providers to record spontaneous video answers and automatically saves these files to the cloud.

In this report I:

- Outline the design of the system,
- Explain the functionality of each page,
- Describe how I went about implementing each page,
- Describe the testing undertaken and the actions arising from this testing,
- And deliver some reflections on how the project went, what I have learned, and how I will apply this in future.

System design

Channels

The video answers are categorised into "channels" of people who provide:

- Domestic support: cleaners, babysitters, drain cleaners;
- Education: tutors of various kinds;
- Home improvement: builders, electricians, carpenters, architects;
- Legal support: solicitors, barristers;
- **Mental health: psychotherapists, coaches, counsellors; and**
- Physical health: physiotherapists, dietitians, massage therapists.

As previously mentioned, this project focusses on the mental health channel but is designed to be extensible to the other channels.

Users

The site has a hierarchy of users who have progressively more permissions.

Casual viewers	Signed-in users	Signed-in providers	Administrators
Can view video answers in all channels Can browse profile summaries Can ask questions	As for a casual viewer plus: Can like video answers and providers Can contact providers Can manage their own details	As for a signed-in user plus: Associated with a single channel Can create video answers Can manage their profile and video answers	Can add and remove questions Can add and remove users and providers

A note on "Providers" vs "Posters"

I have grappled with what to call the different kinds of users on the site. I was originally thinking "questioner" and "answerer" but that is a mouthful. Then I thought around "video watcher" and "video poster". Latterly I have been calling the people recording the videos "providers" (of services) as user testing indicated that this was a more natural word than "poster". Subsequently I have changed to "provider" across the user interface of the site. In the code and the more technical parts of this report, you will see many mentions of "posters". I hope you will now understand that the words "poster" and "provider" are interchangeable.

Choice of stack

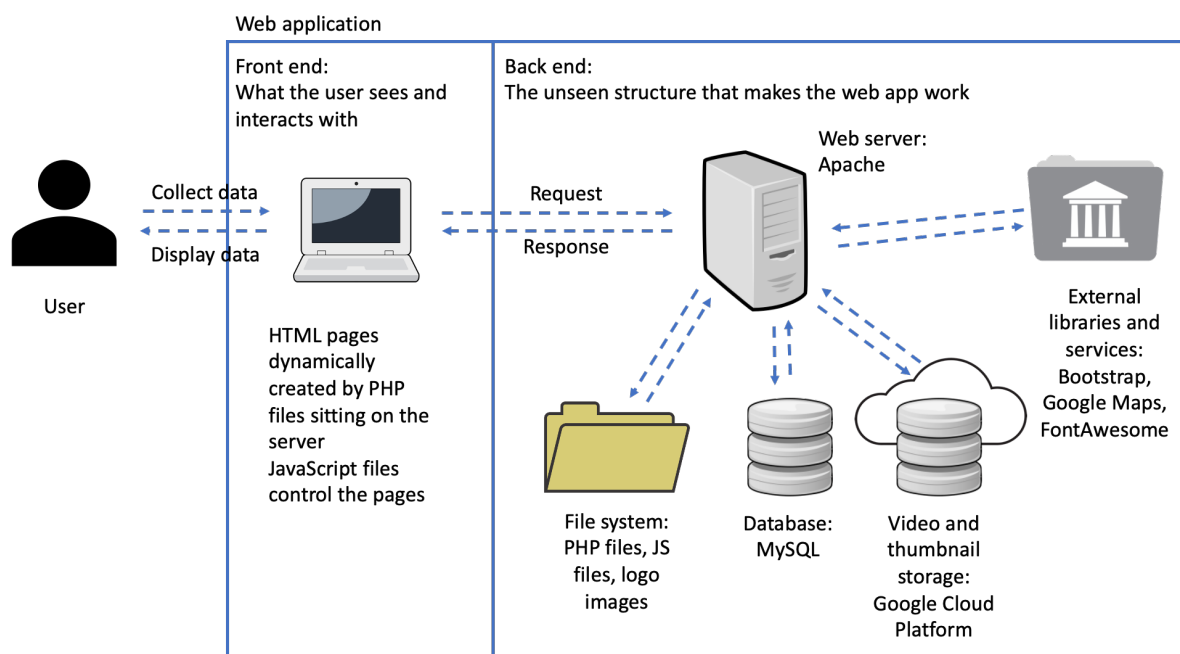
In my project proposal (Evans-Lacey, 2022) I outlined 2 alternative tech stacks: MongoDB, Express.js, React.js, Node.js (MERN) and Linus, Apache HTTP Server, MySQL, PHP (LAMP). Coming into the project my attitude was bullish: I would use this as an opportunity to learn and implement my website using what I considered to be the modern approach: MERN.

During the first two weeks of the project I immersed myself into learning Node.js, Express.js, and react.js. For Node.js and Express.js I studied an 8 hour YouTube course from FreeCodeCamp (Smilga, 2021), for React I studied their 10 hour course (Smilga, 2020). While the courses were excellent they were took me longer than I expected to work through.

In the third week I reflected on my progress and assessed that it would take another 2 weeks of building case studies (Smilga, 2021) before I would be ready to start creating my own content. Even then there was a risk that I may become stuck with the new technology. I considered 2 options: either find significant external support and continue down the MERN route, or use the LAMP stack. I chose the latter.

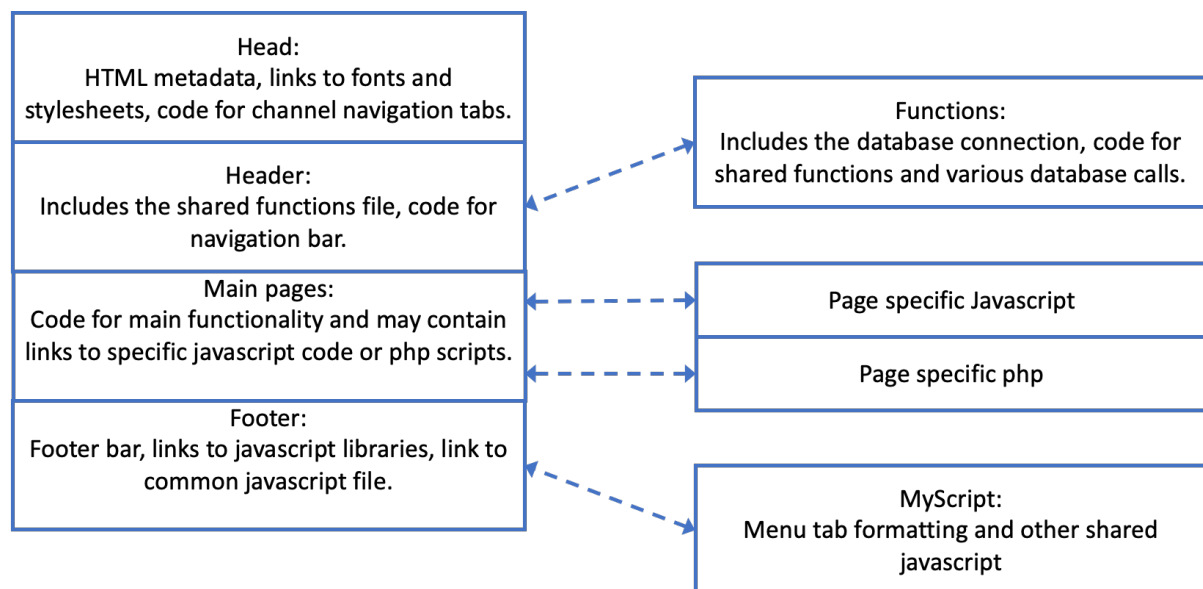
The third week was devoted to reacquainting myself with HTML, CSS, JS, MySQL, PHP, (Traversy, 2022) and the formatting library Bootstrap (Bootstrap, n.d.), all languages that I have only a little previous experience of, and that I haven't touched for 2 years.

High level structure



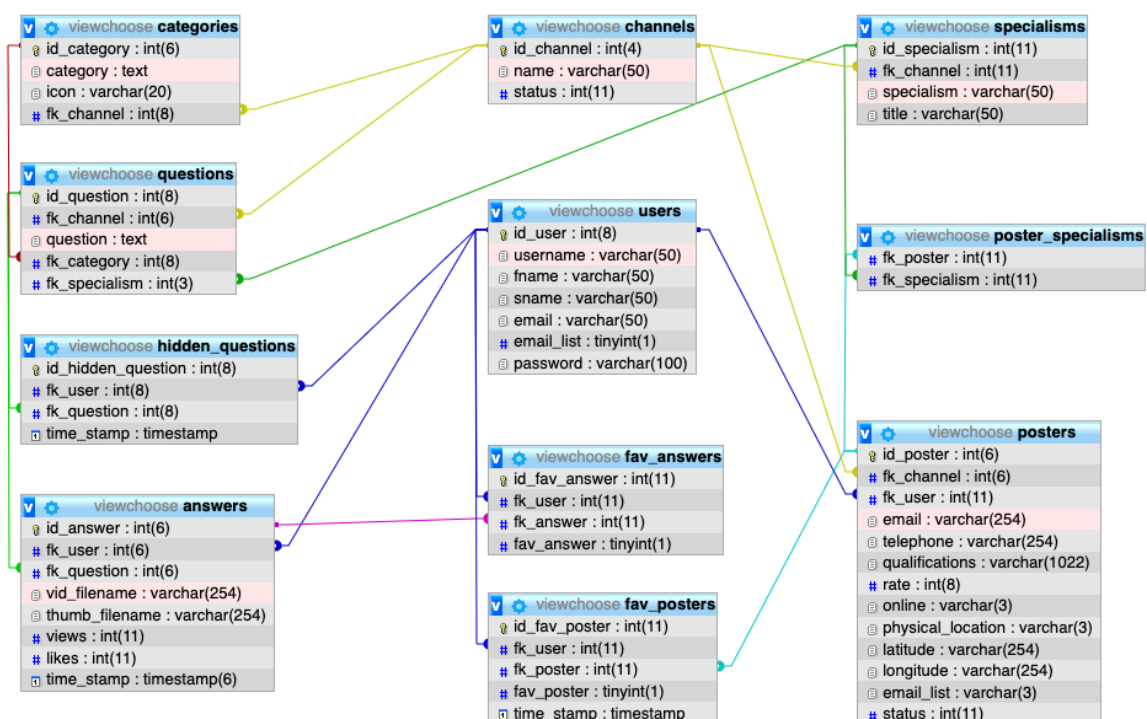
The website pages are created with PHP. Some of the HTML files created by the PHP are heavily controlled by JavaScript files which control what the user sees on the screen. Behind the scenes there are various worker files, especially those interacting with the database. Formatting and some other features are handled by external libraries, and the video files and thumbnails are all saved into the cloud.

Page structure



Most of the pages in the site utilise shared components to reduce replicated code. The particular main page will "include" a common header and footer. The header includes a functions file and a head file. The head file contains all the links to formatting cascading style sheets (CSS) from the Bootstrap and font libraries and a site specific CSS file that can be used if any overriding is required. A number of pages include extra PHP worker files to execute certain actions such as saving to or retrieving from the database. The pages that record and play the videos utilise page specific JavaScript files to control their operation when loaded.

Relational database model



The database model is built in MySQL. Every user has a single entry in the "users" table containing a minimal amount of contact information and a hashed password for logging-in. Each channel (e.g. Mental Health) has a number of question categories associated with it (e.g. about the provider) in the "categories" table. Each category has a number of questions associated with it (e.g. "How did you become interested in mental health") stored in the "questions" table. Each question has a number of answers by different providers associated with it. These answers are in video and thumbnail form and are stored in a Google Cloud Storage bucket. Links to these files are in the "answers" table. Each user may have a number of liked (or disliked) providers or videos. These are stored in the "fav_" tables. Each user may be a provider of services in none, one, or many channels. Each of their provider profiles are stored in the "posters" table. Each provider / poster may have none, one, or many specialisms. A list of the specialisms is stored in the "specialisms" table and these are linked to the providers in the "poster_specialisms" table. Each poster will have many video answers but may have chosen to hide certain questions rather than answering them. These hidden questions are stored in the "hidden_questions" table.

File structure

The following table gives a summary of the functionality of each of the files:

Folder	File	Function
css	Bootstrap and site-specific CSS files	Formatting
images	Various image and video files	Site branding and introductory video content
includes	db-inc.php	Connects to the database
	functions-inc.php	Various shared functions esp to access the database
	login-inc.php	Checks login information and sets session variables
	logout-inc.php	Unsets session variables
	save-fav-answer-inc.php & save-fav-poster-inc.php	Save changes in favourite status
	save-hidden-question-inc.php	Saves a question a provider does not want to answer
	save-video-file-inc (local).php & save-video-file-inc.php	Saves video files either locally or to Google Cloud Platform
	signup-poster-1-inc.php & signup-user-inc.php	Validates signup information and saves it to the database
js	myScript.js	Shared JavaScript including setting the formatting of the menu bars
	player.js	Page specific JavaScript to play videos, save favourites, and manage the video list

Folder	File	Function
	recorder.js	Page specific JavaScript to control the asking of questions and the capture of videos and thumbnails
love	Various google client secrets	Accessing online services
uploads	Various .mp4 and .png files	The folder I used when I was saving files locally
vendor	Various library files including Google	Local code required to run various libraries
root	.gitignore	Folders not to be uploaded to GitHub: love, uploads, and vendor
	composer.json & composer.lock	Locks the dependencies of the project to a known state
	One .php file for each of the web pages	Contains the main PHP and HTML for each of the web pages

Workspace setup

```

1  <?php
2  // ob_start(); // This stops the error message that the headers are already set
3  include_once "head.php";
4  include_once "includes/functions-inc.php";
5
6  $_SESSION['id_channel'] = 0;
7  echo "<SCRIPT> var channel = " . $_SESSION['id_channel'] . "</SCRIPT>";
8  ?>
9
10 <main>
11 <div style="background-color:crimson">
12 <div class="container">
13 
14 </div>
15 </div>
16 <div class="container my-3">
17 <div class="row my-5">
18 <div class="col-3">
19 <?php if(isset($_SESSION['fname'])) {
20     echo "<h3 class='text-center'>Hi " . $_SESSION['fname'] . ", let's find a professional you
21     vibe with. Choose a channel from the tabs on the top menu.</h3>";
22 } else {
23     echo "<h3 class='text-center'>Let us help you find a professional you vibe with.</h3>";
24 } ?>
25 </div>
26 <div class="row text-center">
27 <div class="col-3"><p class="fas fa-video fa-4x"></p><p>View video answers to questions that

```

I chose Visual Studio Code (VS Code) (Visual Studio Code, n.d.) as my Integrated Development Environment (IDE) as this was the IDE used in the Cloud Computing module of the MSc. To develop using the LAMP stack I needed a local Apache server with PHP and MySQL. To facilitate this I installed a web development solution called MAMP (MAMP, n.d.).

In order to have the results of my code dynamically appear in my browser when I saved it in the IDE I installed the following extensions: Live Server (Dey, n.d.), PHP Server (brapifra, n.d.), and PHP Intelephense (Mewburn, n.d.) in addition to the Live Server extension for the Chrome browser (Dey, n.d.).

In order to see PHP errors I set "display_errors = On" in the MAMP PHP configuration initialisation file php.ini (kn000x, 2014).

All development work was completed using a Google Chrome browser with developer tools.

Page design

Navigation

Channel tabs (logged-out)

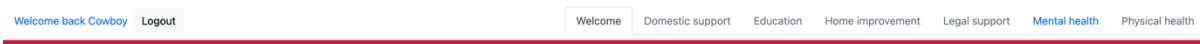
When logged out as a user:



The channel navigation is common across the site and allows users to sign-in, log-out or access the different categories quickly.

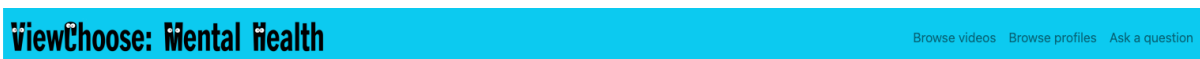
Channel tabs (logged-in and signed-up as a provider)

When logged in as a user:



When the user is logged in the left hand side of the channel tabs welcomes them back and gives them the option to log-out.

Navigation bar (logged-out)



The navigation bar is common across all pages within a channel (but not the welcome page). On the left we have channel specific branding (in this case mental health) which, when clicked, brings the user back to the "Browse videos" page. When logged-out the link options are limited to browsing videos, browsing profiles, and asking questions.

Navigation bar (logged-in and signed-up as a provider)



When logged-in as a provider the user is able to record and manage video content.

Responsiveness



Many users will want to use the site on a mobile device. If they did this with the standard layout many of the features would become very difficult to use. (The Medium Well, n.d.)

The menu is responsive such that the screen is narrow then the menu collapses into a "burger bun" which reveals a list of menu items when clicked.

Footer



The footer is common across the pages though its colour changes accordingly. In addition to the current social media links, it could hold links to less often parts of the site such as legal small-print.


Welcome


[Login](#) [Sign-up](#)


Welcome Domestic support Education Home improvement Legal support **Mental health** Physical health


ViewChoose

Let us help you find a professional you vibe with.



View video answers to questions that interest you


Like your favourites


Get to know your favourites



Choose a professional you vibe with

What is ViewChoose?




0:00 / 0:02

Who are your providers?



0:00 / 0:05


How do I find a provider I vibe with?



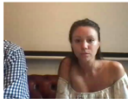
0:00 / 0:04

Or click on a video that interests you:


What is anxiety?




What causes depression?




What is the difference between psychotherapy and counselling?



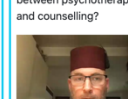
What is the difference between psychotherapy and counselling?




What is anxiety?




What is the difference between psychotherapy and counselling?



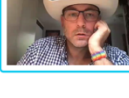
How did you become interested in mental health?




Where do you go to relax?




How does hypnosis work?



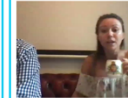
Who invented NLP?

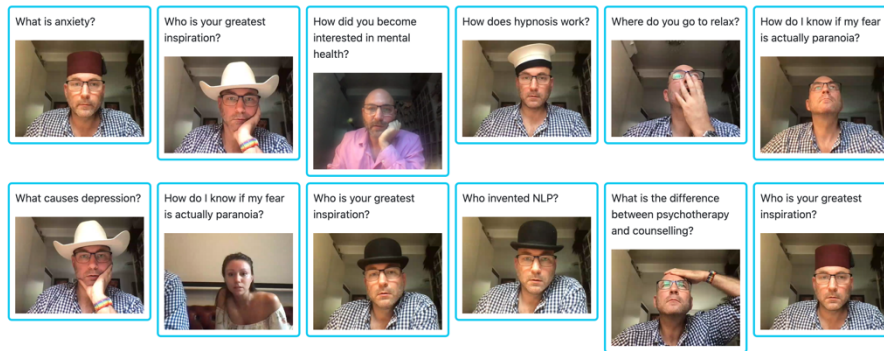


What is the difference between psychotherapy and counselling?



How did you become interested in mental health?





© Richard Evans-Lacey Sep 2022

The welcome page is intended to introduce users to the site and guide them quickly into useful content. It consists of a branded banner, an iconic summary of the process, a video introduction, and some video thumbnails.

Branded banner



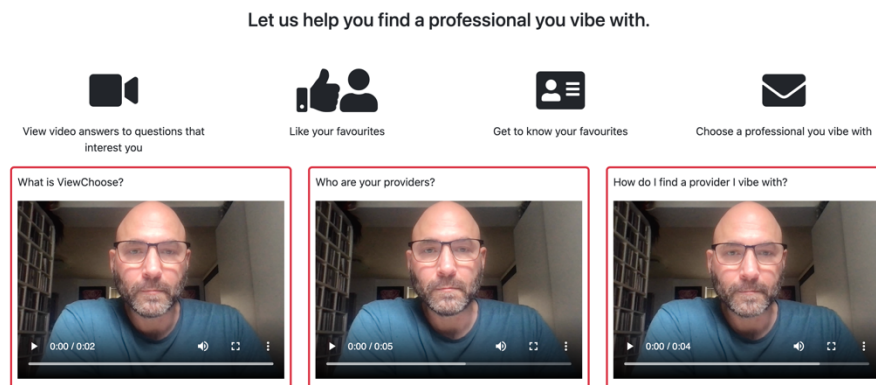
I needed a brand that was general enough to sit across channels, relevant to the "finding via video" nature of the site, and had an available URL. "ViewChoose.com" seemed to fit the bill.

The eyeball graphic design is intended to emphasise the "watching videos" nature of the site and also to introduce a friendly personality to an otherwise potentially dry activity.

"ViewChoose" sounds somewhat like "YouTube" and the red colour and font style are playful nods towards YouTube's logo (Logos-World, 2017):



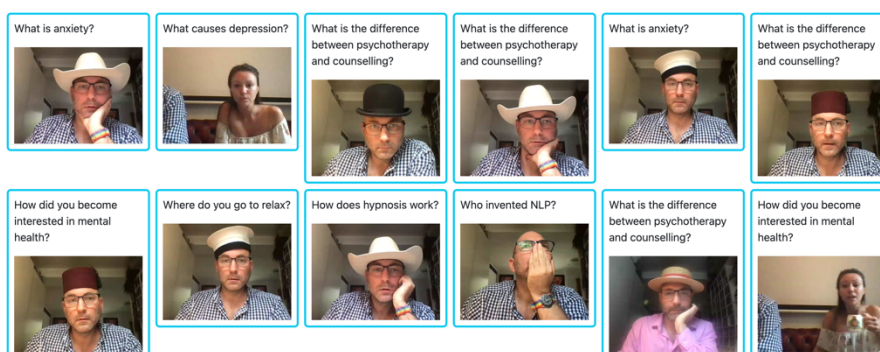
Iconic summary + video introduction



The intention of this section is to introduce the viewer to the concept of the site. The logos used here are repeated on the "Browse videos" page of the site.

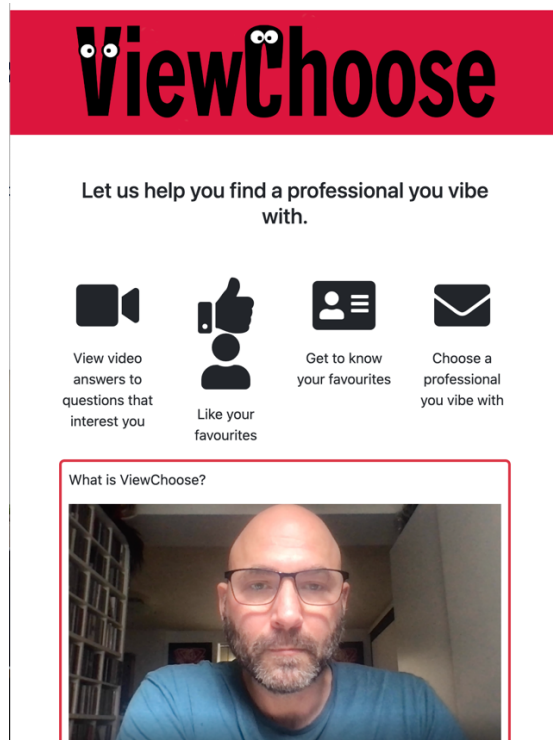
Video thumbnails

Or click on a video that interests you:



The intention of adding the video thumbnails is to give the viewer an idea of the wide variety of information that is available across the site. In this example the different hats represent different providers. Note that all the outlines are light blue as this is the logo colour for mental health providers. In a site with multiple channels each video would be outlined by that channel's logo colour.

Responsiveness



When this page (and all the others that follow) are viewed on a narrow device such as a mobile phone the screen elements automatically change from being displayed next to each other to being displayed stacked on top of each other.

Browse videos

Username Password Login Sign-up

Welcome Domestic support Education Home improvement Legal support **Mental health** Physical health

ViewChoose: Mental Health Browse profiles Ask a question

Keyword search:


Show me videos about:

[To do] Show videos from specialists in:

Who offer:
☒ online consultations
☐ [To do] in-person consultations

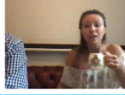
Apply filter

Hey Businessman, who invented NLP?




0:02 / 0:02

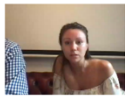
How did you become interested in mental health?




Who is your greatest inspiration?




How do I know if my fear is actually paranoia?




What is the difference between psychotherapy and counselling?




What is the difference between psychotherapy and counselling?




Who is your greatest inspiration?




How do I know if my fear is actually paranoia?



Where do you go to relax?



Who is your greatest inspiration?



The play videos page is the centre of the website. It enables a view to watch video answers to questions that interest them. There are three main parts of this page: the filters, the watch screen, and the video queue.

Filters

Keyword search:

Show me videos about:

[To do] Show videos from specialists in:

Who offer:
☒ online consultations
☐ [To do] in-person consultations

Apply filter

Show me videos about:

- ✓ All
- Approaches and techniques
- Being a client
- Helping others
- Morality and ethics
- Psychological issues
- Psychological theory
- Relationships
- Self help
- The provider
- The provider's style

[To do] Show videos from specialists in:

- ✓ All
- Coaching
- Counselling
- Healing
- Hypnotherapy
- Psychiatry
- Psychology
- Psychotherapy

Apply filter

The filters allow the user to restrict the videos shown according to keywords in the video title, video categories, the particular specialism of the provider, and whether the provider provides online or in-person consultations.

The keyword filter restricts the videos shown to those with titles (the questions the provider is answering) containing the keyword chosen. In the event that more than one word is input then only the first is searched for.

The dropdown menus allow the user to search for questions that fall within a certain category and for providers who have a particular specialism.

The online-consultations tick-box allows users to choose to only watch videos from providers who they can then book an online session with.

The in-person consultations tick allows users to choose to only watch videos from providers within a certain radius of a location chosen by the user.

Watch screen

Hey Rich, how did you become interested in mental health?



The watch screen is where the videos from the queue are played. At the top is the question that the provider is answering. The main screen has standard play / pause control, a volume button, a full-screen button, and options to download the video or go picture in picture.

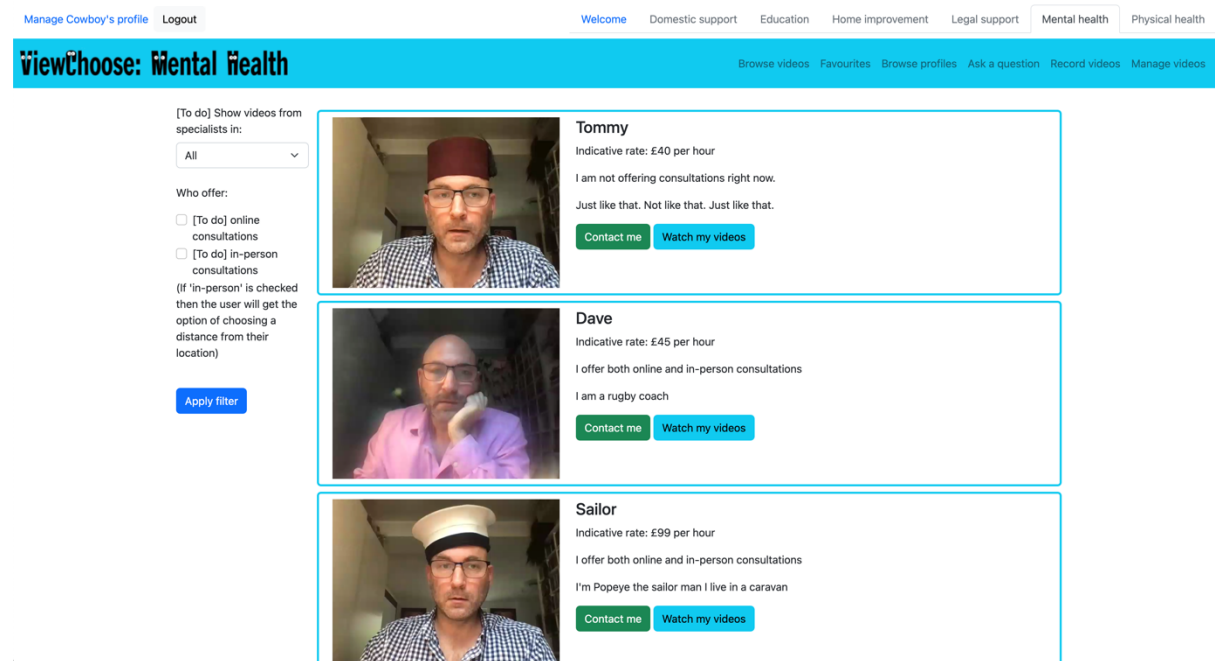
Under the screen are buttons for skipping backwards and forwards in the queue, disliking or liking the video or the provider, and a link to the provider's profile and contact page.

Video queue

The video queue is a random selection of videos that have been filtered according to the viewer's wishes that is loaded into the page every time it is refreshed.

In the event that the user dislikes a video or a provider then that video stays in the queue until the page is refreshed, then the particular video or provider is excluded from showing up in any further searches.

Browse profiles



The browse profiles page gives the user an alternative search experience more akin to a standard listing site. It consists of filters and a list of provider profile pictures and summary information (excluding any providers that are on the user's "dislike" list).

The "Contact me" button takes the user to the provider's profile page. The "Watch my videos" button takes the user to the "Browse videos" screen which is then populated with a queue of videos from this provider.

Ask a question

[Manage Cowboy's profile](#) [Logout](#) [Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)

ViewChoose: Mental Health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)

Ask a question

Do you have a question that you would like our providers to answer? Please use the form below to let us know.

(You can use the same form if you have any other feedback on the site: it all gets handled by the same real humans.)




Your name:

Your email address:

, Message: (You have 1000 characters remaining)

Your question(s) should all start with 'Who', 'What', 'Where', 'When', 'Why', or 'How'.

Send

© Richard Evans-Lacey Sep 2022

If the user has a suggestion for a question that should be included in a channel or a message for the managers of the site this is where they can communicate it. Once received the questions will be manually considered prior to inclusion to ensure that there are no existing close alternatives and that they are correctly phrased.

User sign-up

Sign-up to ViewChoose

Great to have you with us! Soon you will be able to save your favourites and contact providers directly.

User ID (this could be your email)

First name

Second name

Email

Password

Repeat password

[Read the small print here](#)

☐ I have read the small print and agree to it all.

☐ I would love you to email me about new videos and other interesting stuff from time to time.

The sign-up form is pretty self-explanatory. To ensure it the form is correctly filled in all the fields have full error checking.

Sign-up to ViewChoose

Great to have you with us! Soon you will be able to save your favourites and contact providers directly.

User ID (this could be your email)

Please complete.

First name

Please complete.

Second name

Please complete.

Email

Please enter a valid email.

Password

Please ensure passwords match.

Repeat password

Please ensure passwords match.

[\[To do\] Read the small print here](#)

☐ I have read the small print and agree to it all.

Please read and tick.

☐ I would love you to email me about new videos and other interesting stuff from time to time.

Submit


Favourites

[Manage Cowboy's profile](#) [Logout](#) [Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)

ViewChoose: Mental Health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)


Your favourite providers

Rich




[Remove](#)

Businessman



[Remove](#)


Cowboy



[Remove](#)


Your favourite video answers

Who invented NLP?




[Remove](#)

How does hypnosis work?




[Remove](#)

What is the difference between psychotherapy and counselling?



[Remove](#)



© Richard Evans-Lacey Sep 2022


If the user is signed-in then any favourite providers or video answers may be re-accessed via this page. Clicking on a favourite provider takes the user to the provider's contact page. Clicking on a favourite video takes the user to watch the video on the "Browse videos" page.

Provider contact


[Manage Cowboy's profile](#) [Logout](#) [Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)

ViewChoose: Mental Health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)

Hi, my name is Cowboy



Indicative rate: £48 per hour
I offer both online and in-person consultations
I ride horses



Please drop me a line if you think I may be able to help

From: Cowboy , Email: cowboy@mail.com , Message: (You have 1000 characters remaining)

Hi Cowboy,

I've watched your videos and I like your vibe.

Here is a bit of information about me and what I'm looking for:

Please email me back or call me on:

Send

How does hypnosis work?

Who is your greatest inspiration?

What is the difference between psychotherapy and counselling?

What causes depression?

How do I know if my fear is actually paranoia?

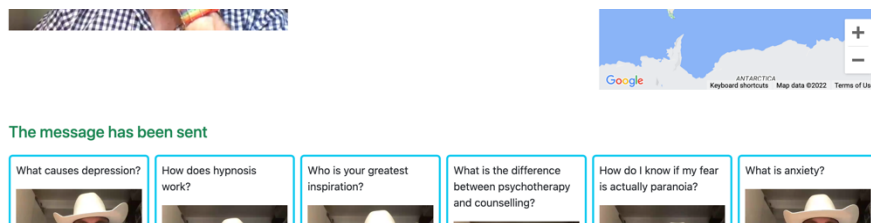
What is anxiety?

© Richard Evans-Lacey Sep 2022

The provider contact page is in three parts: their profile information, a contact form, and a library of their videos.

The profile information summarises the provider's qualifications, how they practice, and an indicative fee. If the provider practices in-person then it also provides a map of their location.

The contact form is pre-populated with the signed-in user's name and email address. The freeform text box suggests a message format without requiring the user to use that format. It has a maximum message length of 1000 characters which are dynamically tracked. When the message is sent the message box is replaced by the confirmation "This message has been sent":



The library contains all the videos from this particular provider. When the user clicks on a video they are taken to the "Browse videos" with a queue of videos exclusively from this provider:

viewChoose: Mental Health
[Browse videos](#)
[Favourites](#)
[Browse profiles](#)
[Ask a question](#)
[Record videos](#)
[Manage videos](#)

Keyword search:

Show me videos about:

All

[To do] Show videos from specialists in:

All

Who offer:

☒ online consultations
 ☐ [To do] in-person consultations

Apply filter

Hey Cowboy, who is your greatest inspiration?

0:01 / 0:02

How does hypnosis work?

How do I know if my fear is actually paranoia?

What causes depression?

What is the difference between psychotherapy and counselling?

Who is your greatest inspiration?

What is anxiety?

Provider sign-up

[Manage Rich's profile](#)
[Logout](#)
[Welcome](#)
[Domestic support](#)
[Education](#)
[Home improvement](#)
[Legal support](#)
[Mental health](#)
[Physical health](#)

viewChoose: Mental Health
[Browse videos](#)
[Favourites](#)
[Browse profiles](#)
[Ask a question](#)
[Record videos](#)
[Manage videos](#)

Update poster details

Hi Rich 19. Thank you for keeping your details up to date.

Professional email

You can use your user email or a different one.

richard@psychicplumbing.com

Telephone number

Please enter a telephone number (including international dialing code) we can use to contact you. This will not be published on your profile.

+442073771918

Specialisms

Which approaches do you consider yourself qualified to deliver?

☐ Coaching
 ☐ Counselling
 ☐ Healing
 ☐ Hypnotherapy
 ☐ Psychiatry
 ☐ Psychology
 ☐ Psychotherapy

Qualifications and memberships

Please describe your qualifications and any professional memberships. You have 1000 characters remaining.

I know lots of stuff
I have a MEng
I almost have an MSc

Indicative rate

What is your standard rate? You can explain any concessions or block booking deals when you are contacted.

£ 81 per 60 minutes

Your rate is £81 per hour.

Checking the 'online consultations' box will allow you to work with clients from around the world.

☒ I offer online consultations

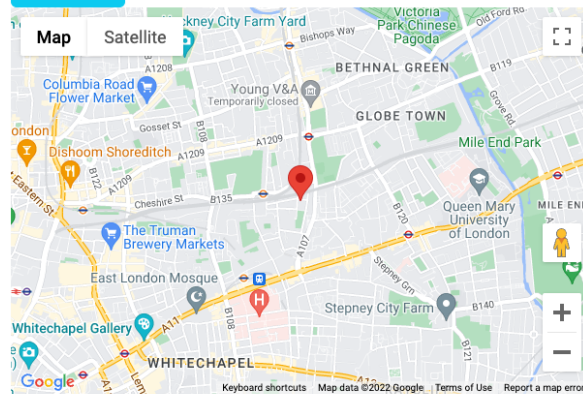
Checking the 'in-person consultations' box will allow you to work with clients in person.

☒ I offer in-person consultations

Please tell us the postcode or address of your main practice location so we can locate it on the map.

Postcode / address

Find location



Your location is lat: 51.5237456, lng: -0.05626159999999999.

[Read the small print here](#)

☐ I have read the small print and agree to it all.

☐ I would love you to email me about new questions and other interesting stuff for mental health professionals (recommended).

Sign up



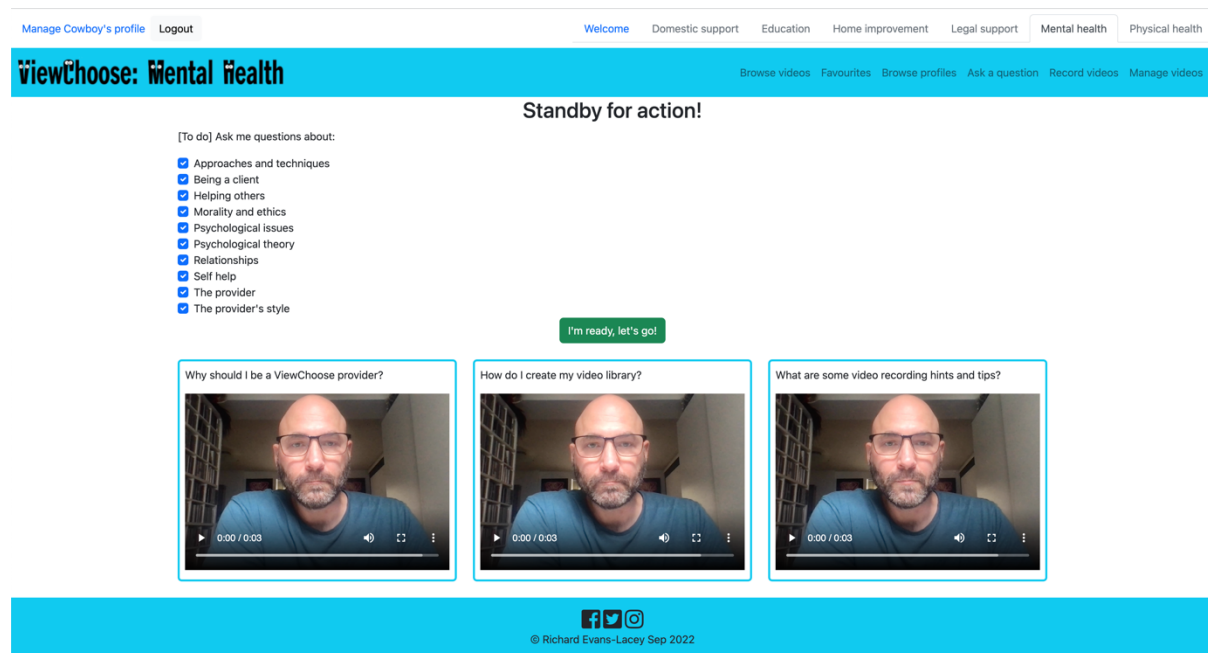
© Richard Evans-Lacey Sep 2022

If a user wants to record videos, they need to become a provider. This is the form they use to sign-up. It has full error checking and a number of other features:

- People often have a separate work email address for different line of work so this form gives the user the option of adding an email that is different to their user sign-up form.
- The user can choose one or more specialisms.
- The qualifications and memberships page is limited to a fixed number of characters and features a dynamic character count so the user knows how many characters they have left.

- Different providers will charge in different ways. One may charge £60 for a 50-minute session, another may charge £500 for a block of 6 with the 7th one for free. In order to give users an easily comparable guideline the form dynamically calculates an indicative hourly rate for each provider.
- When a provider chooses the "in-person consultations" option, the form opens up a Google map section. The user is invited to enter their address and the site finds it on a map of the world. The latitude and longitude of the location is stored in the database for proximity calculations.

Record videos intro

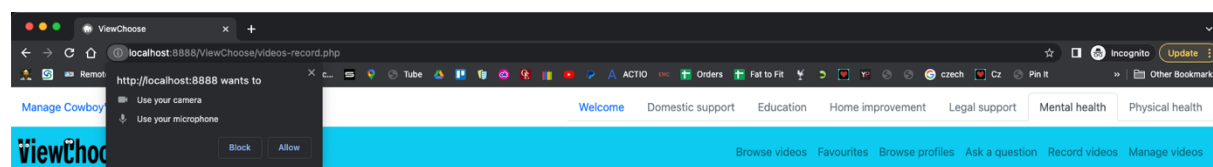


Prior to recording the provider will need to get themselves ready. This page provides motivation for embarking on the recording process (which may be stressful for some) and provides some tips on how to make good quality videos.

The provider can choose between answering randomly chosen questions from across all the categories or narrowing it down to a few. When they are ready to record they press the "I'm ready, let's go!" button and are taken to the "Record videos" page.

Record videos

Accessing the camera and microphone



The first time a user records a video they are asked for permission to access their camera and microphone. From then on, the page goes into automatic recording mode. It has three states: before recording, during recording, and after recording.

Before recording

[Manage Cowboy's profile](#) [Logout](#)

[Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)


viewChoose: Mental health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)

user: 29, id_question: 13

Where do you go to relax?

[Hide, next question >>](#) [Answer later, next question >>](#)

Recording will start in 1



[f](#) [t](#) [i](#) [g](#)

© Richard Evans-Lacey Sep 2022

The provider is asked a question at random from a question "pool". They have 3 seconds to make a choice: If they don't like the question they can choose to hide it, they will never see the question again and the page asks them another question. If they don't want to answer the question right now, they can press the "Answer later" button, the question will be returned to the question pool, and the page asks them another question. If they feel like having a go at answering the question, they simply wait for the 3-2-1 countdown to end and the page starts recording.

During recording

[Manage Cowboy's profile](#) [Logout](#) [Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)


viewChoose: Mental Health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)

user: 29, id_question: 13

Where do you go to relax?

[I'm done!](#)

24



[f](#) [t](#) [i](#) [g](#)
© Richard Evans-Lacey Sep 2022

The page shows it is recording by turning the border of the preview box red. Above the preview box is count-up (less stressful than a count-down) timer, above that the question continues to be displayed. The provider records their answer. To warn them that they are coming to the end of the 60 seconds maximum time the border around the preview flashes red and black. At 60 seconds the recording is automatically stopped. If the provider finishes their answer before then they can press the "I'm done" button to stop recording.

After recording

[Manage Cowboy's profile](#) [Logout](#)

[Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)


viewChoose: Mental Health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)

user: 29, id_question: 13

Where do you go to relax?

[Download video](#) [Discard, next question >>](#) [Save, next question >>](#)

Please review your video below. Pause it to choose your video thumbnail.



[Facebook](#) [Twitter](#) [Instagram](#)

© Richard Evans-Lacey Sep 2022

When recording the video is available for the provider to watch back. The provider has another choice to make: to discard the video or to save it. If they wish to discard the video then the question is moved back into the pool. If they save then a record of the answer is saved in the database and the video and a thumbnail files are saved to the cloud. The video is immediately available in the main site and on the provider's profile. If they would prefer to choose a different thumbnail they can pause the video preview at any point and that is the screenshot that is used.

Manage videos

[Manage Cowboy's profile](#) [Logout](#)


[Welcome](#) [Domestic support](#) [Education](#) [Home improvement](#) [Legal support](#) [Mental health](#) [Physical health](#)

viewChoose: Mental Health [Browse videos](#) [Favourites](#) [Browse profiles](#) [Ask a question](#) [Record videos](#) [Manage videos](#)

Manage videos

Use this page to review your videos and delete any you are no longer happy with. Please note that once a video has been deleted that action cannot be undone.


How does hypnosis work?



Views: #, Likes: #, Dislikes: #

[Clip code](#) [Delete](#)


How do I know if my fear is actually paranoia?



Views: #, Likes: #, Dislikes: #

[Clip code](#) [Delete](#)


What causes depression?



Views: #, Likes: #, Dislikes: #

[Clip code](#) [Delete](#)


What is anxiety?



Views: #, Likes: #, Dislikes: #

[Clip code](#) [Delete](#)


Who is your greatest inspiration?



Views: #, Likes: #, Dislikes: #

[Clip code](#) [Delete](#)

What is the difference between psychotherapy and counselling?



Views: #, Likes: #, Dislikes: #

[Clip code](#) [Delete](#)

[Facebook](#) [Twitter](#) [Instagram](#)
© Richard Evans-Lacey Sep 2022

localhost:8888/viewChoose/videos-play.php

From time to time the provider may wish to review their video library. This page contains a list of videos and details of how many times they have been viewed and liked. If the provider wishes to delete a video and return the question to the question pool they can do that here. If they would like to clip some code to insert the video into their own website then they can do that here too.

Page implementation

Navigation

This consists of login options and two sets of navigation: the channel tabs and the channel specific menus.

Login / logout / sign-up

Session variables are variables that are stored in the server for the duration of a user's visit to the site regardless of the web page that the user is on. The state of the login area is determined by checking for a session variable "id_user" that indicates if there is a user logged in, and "#.id_poster" (where # is a number) that indicates if that user is a provider on channel #.

The user logs in by entering data into a form. When submitted this form sends that data to a file called "login-inc.php". The functionality of this page includes code to check the password that has been entered correctly. For security reasons I only store encrypted passwords in the database so I use an inbuilt PHP function called password_verify() that compares the hashed password that has been saved in the database using the function password_hash(). (php, n.d.), (Krossing, 2018):

```
$passwordCheck = password_verify($password, $row[ 'password' ] );
```

If the password is correct this page sets a number of session variables that identify the user and are used elsewhere in the site. If the user chooses to log-out at any time all of these session variables are unset and the session itself is destroyed by the file "logout-inc.php" (com, n.d.):

```
session_unset();  
session_destroy();
```

Channel tabs

The channel tabs are created utilising the "nav-tabs" bootstrap class on an unordered list of items (Bootstrap, n.d.):

```
<ul id="menuTabs" class="nav nav-tabs ml-auto">  
  <li class="nav-item"><a id="tab0" class="nav-link"  
href="index.php">Welcome</a></li>  
  <li class="nav-item"><a class="nav-link disabled"  
href="">Domestic support</a></li>
```

To set the tab that the user is currently on it must be "active". This is done in the myScript.js page based on the channel variable set on the "header.php" page:

```
switch (channel) {
    case 1:

document.getElementById('tab0').classList.remove('active');
    document.getElementById('tab1').classList.add('active');
```

Navigation bar

The navigation bar is not shown on the index page, only the channel pages. The channel can be set by calling any page with an appropriate `id_channel` GET variable in the URL, for example "videos-play.php?id_channel=1". The header file checks for this GET variable, sets it as a session variable, and also as a JavaScript variable so that the navigation can be appropriately formatted:

```
if (isset($_GET['id_channel'])) {
    $_SESSION['id_channel'] = $_GET['id_channel'];
}
echo "<SCRIPT> var channel = " . $_SESSION['id_channel'] .
"</SCRIPT>";
```

Depending upon the "id_user" and "#.id_poster" session variables, appropriate levels of menu options are displayed.

Responsiveness

When the screen is narrow then a "burger bun" button is shown and the content in the "navbarToggler" div is displayed within a drop-down menu. The code to do this makes use of the Bootstrap collapse plugin (Bootstrap, n.d.):

```
<button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarToggler" aria-
controls="navbarToggler" aria-expanded="false" aria-
label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarToggler">
```

Footer

The colour of the footer is also changed by the function in "myScript.js". The social media icons are sourced from a library called "font awesome" (Font Awesome, n.d.) and accessed using simple code:

```
<i class="fab fa-facebook-square fa-2x"></i>
```

Welcome

The welcome page is "index.php", the default landing page. I call this page channel 0 for the purposes of formatting the tabs and the footer.

This is a relatively simple page that displays some introductory videos and some thumbnail videos. To display the thumbnails it queries the database with the function "fetch_answers()" then loops over the answers creating HTML code for each answer. The filenames of the thumbnails are stored in the database and the thumbnails are stored in a Google Cloud Platform bucket called "viewchoose". The images are simple to access by setting the image source to a filepath concatenated with the filename:

```
$file_path = "https://storage.googleapis.com/viewchoose/";
<img class="img-fluid" src=<?php echo $file_path .
$answer["thumb_filename"]?> alt=" A thumbnail picture for the
answer video
">
```

fetch_answers()

This function returns an array of answer data from the database and is used on multiple pages. It pulls together data from multiple tables and filters it according to multiple parameters:

```
function fetch_answers($id_channel = "all", $like = '',
$id_category = 'all', $online = 'all', $id_answer = "all",
$id_poster = "all", $fav_answer = "all", $fav_poster = "all",
$limit = 96)
```

This function returns all the answers in the database according to the following Structured Query Language (SQL) query:

```
$sql = "SELECT * FROM questions AS q, answers AS a, users AS
u, categories AS c, posters AS p WHERE q.id_question =
a.fk_question AND u.id_user = p.fk_user AND u.id_user =
a.fk_user AND q.fk_category = c.id_category $id_channel
$id_category $online $id_answer $id_poster $like $fav_answer
$fav_poster ORDER BY RAND() LIMIT $limit";
```

Notice that the query contains the parameters listed in the function, however, with the exception of \$limit, the parameter defaults won't actually work within the query. There is an intermediate step where each parameter is transformed into an appropriate line of SQL. For example, to filter for only the answers from a given channel the intermediate code is:

```
if ($id_channel == "all") {
    $id_channel = "";
} else {
    $id_channel = "AND q.fk_channel =" . $id_channel;
}
```

If the channel is set to the default "all" then no extra SQL is inserted into the query, however, if the channel is set to a number then the query only returns answers where the fk_channel column of the question table is equal to that channel number.

The most complex part of this intermediate code is the part that filters according to whether the signed-in user has or hasn't set the particular answer as a favourite. For example, if we only want answers from posters who have been selected as a favourite (\$fav_poster == 1) then the SQL code to be inserted needs to check if the id of that poster appears IN the "fav_answers" table of the database for that user. It achieves this by using a nested SELECT statement:

```
$fav_poster = "AND a.id_poster IN (SELECT fa.fk_poster FROM
fav_posters AS fp WHERE fp.fk_user = " . $_SESSION['id_user']
. " AND fp.fav_poster = " . $fav_poster . ")";
```

Responsiveness

I made my design responsive by utilising Bootstrap's "grid" system (Bootstrap, n.d.). The code to do this has a "container" division that contains "row" divisions. Each row contains 12 units of width and can be divided into columns containing units of width that total to 12. In the following case the column is one of 4 across the row:

```
<div class="container">
  <div class="row">
    <div class="col-3">
```

Browse videos

The HTML code for the watching videos is located in "videos-play.php" which is controlled by "player.js". The page is divided into the filters, the watch screen, and the video queue.

Filters

The filters are a set of controls on a form that the user can select from that restrict the set of answers that are pulled into the queue. When the form is submitted the "videos-play.php" page is re-loaded with the selections passed into itself as GET variables.

Watch screen & Queue

Every time the videos-play.php page is loaded it fetches an array of answers using the function "fetch_answers()" with parameters set according to the GET variables passed in to the page by the filters. If the user is logged in then the page also loads that user's favourite information from the database. In order to access this information in the "player.js" file I encode it into JavaScript Object Notation (JSON) (CodexWorld, 2017):

```
<script>
  var answers = <?php echo json_encode($answers);?>;
</script>
```

The player.js file dynamically updates the title and source of the main "vidPlayer" <video> tag, and also the titles and thumbnails of the items in the queue. The video content is streamed directly from Google Cloud Platform. The first video to play is the video at vidIndex = 0 in the queue with the queued thumbnails offset from there. When one video has finished there is a brief pause and the video and queue are reloaded (this time from an

incremented vidIndex =1). In the event that a user has requested a particular video to play first then the answer information for that video is pushed to the front of the queue prior to the first video being played.

Like / dislike buttons

Coding video and answer likes / dislikes proved to be rather involved. The user's like / dislike information is loaded when the page loads but then changes as soon as the user hits any of the thumbs up or thumbs down buttons. When these buttons are pressed the player.js file runs the function saveFavPoster() or saveFavAnswer() which, in turn, calls the included php files "save-fav-poster-inc.php" or "save-fav-answer-inc.php" to actually write / update / delete the new "like" information to the database.

The problem comes when we want to have accurate button up / button down behaviour on each video in the queue as the database is now out of sync with the data that was downloaded when the page was originally loaded. For example, let's assume I have no favourite providers but then watch a great video by Dave and press the thumbs-up button. As I continue to watch more videos I see another video by Dave but the like button won't show as pressed as I hadn't liked him at the time the page was loaded. Very confusing for a user!

I resolved this issue by creating a temporary array of favourites in the "player.js" file and basing the button up / button down behaviour on that. Now when the user presses "like" it not only updates the database but also updates this temporary table.

Browse profiles

This page is to do due to time constraints and will utilise database calls and presentation similar to that in the profile page.

Ask a question

This page is in the "contact.php" file. It consists of a form to gather data and a function to send that data as an email.

There is some validation on the form itself which I will cover in more detail when I consider the sign-up pages.

This page uses an inbuilt PHP function called mail() (php, n.d.) to send out an email with a descriptive title that the receiver (in this case our ViewChoose inbox) can reply to:

```
$subject = "ViewChoose contact from: ".$fromName;
$headers = "from: ".$fromEmail."\r\n";
mail($toEmail, $subject, $message, $headers);
```

User sign-up

The user sign-up page is designed to validate data prior to saving to the database. When the user submits the form, the data is set to the "signup-user-inc.php" page for checking. The already entered data and any errors are sent back to the user sign-up page so the user can correct them easily.

The form data is sent to the included page by POST variables. As soon as the variables arrive they are cleansed using the function "cleanseInput()" that trims any spaces from the

beginning and end of the input, strips out any back slashes, and prevents the browser from reading any HTML tags such as "<script>".

```
function cleanseInput($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```

To ensure the user only uses letters, spaces, and dashes in their first and second names we use a regular expression match:

```
preg_match("/^[a-zA-Z '\-]*$/", $sname);
```

Persisting data and providing error messages

Any errors and the existing data that the user has entered are appended to the "signup-user.php" header as GET variables (I don't send back the password!) (Krossing, 2018). I found this method a bit messy and decided to use session variables instead for the poster sign-up form.

When data has been sent back to the form it is re-populated into the input box "value" using PHP to echo the appropriate value. The class attribute contains the Bootstrap formatting "form-control" and also a designator "is-invalid" which is echoed by the PHP code if there is an error variable associated with the input:

```
<input type="text" class="form-control <?php echo $errUid ?  
'is-invalid' : '';" name="uid" placeholder="User ID"  
value="<?php echo $uid;">
```

The division containing the error message is a sibling to the input and contains the Bootstrap class "invalid-feedback". As a default this division is hidden, however, if the 'is-invalid' class is set on the input it is set to visible:

```
<div class="invalid-feedback">  
    <?php echo $errUid;">  
</div>
```

If there is an error with the user id then the \$errUid error message is displayed in a nicely formatted box:

User ID (this could be your email)

Please complete.

Securing the inputs from SQL injection attack

SQL injection attack is where a hacker uses forms to insert malicious SQL code that pulls extra data out of the database or corrupts it in some way (Wikipedia, n.d.). I counteracted this by preparing the SQL statement first and then binding parameters to it (W3 Schools, n.d.):

```
mysqli_stmt_prepare($stmt, $sql);  
mysqli_stmt_bind_param($stmt, "ssssss", $username, $fname,  
$sname, $email, $hashedPassword, $emailListBoolean);
```

Favourites

This page uses similar code to the "Browse videos" page in terms of obtaining a filtered list of answers, displaying them, and updating the user's favourites if they want to remove an item.

Provider contact

This page presents all the information associated with a given provider. The map code is similar to that in the "Provider sign-up" page that I will introduce in the next section. The contact form code is similar to that in the "Ask a question" page. The code to display the video library is similar to that on the "Welcome" page. A difference here is that when a video is clicked it sends the provider id along with the answer id to the "Browse videos" page so it only shows videos recorded by this provider.

Provider sign-up

A single user can be a provider in multiple channels, each with its own email address, specialisms, etc. To accommodate for later extension on this page I have used session variables which are currently preceded with a hard coded "1" but could be encoded with the channel number of the current channel.

Much of the code on this page is similar to the user sign-up page, however, on this page I decided to use session variables rather than GET variables to persist the data and pass back any form errors. I found this to be easier to keep track of.

On the form I experimented with a number of validation techniques:

Specialities: check boxes

I created code that pulls the current list of specialities for the particular channel into an array and then looped over that array to create the tick boxes. When the provider chooses a tick box the form creates an array of selections (Agarwal, 2020). This array gets saved into a session variable in order to persist these values on a part-completed form.

Qualifications and memberships: character counter

Rather than trying to maintain a list of potential qualifications I decided this should be a freeform field and that it should be constrained in some way to prevent the poster from writing too much. To help the poster know how much is too much I included a maximum number of characters and a countdown to show how many they have remaining.

I identified the text area "qualText" and the countdown field "qualCount" in the document and wrote some JavaScript to watch for changes in the text and to update the value of the count:

```
var maxChar = 1000;
qualCount.innerHTML = maxChar;
qualText.addEventListener("keyup", function(){
    var char = qualText.value.length;
    qualCount.innerHTML = maxChar - char;
});
```

Rate: calculated input

The form asks the provider to give an indicative hourly rate. As some providers may have sessions that last for more or less than an hour, I created a function that takes the amount in pounds and the time in minutes calculates the hourly rate for them:

```
updateRate = () => {
    rateInt = (pounds.value / (mins.value / 60)).toFixed(0);
    rateInfo.innerHTML = rateInt;
    rateOutput.value = rateInt;
}
```

Physical location: obtain coordinates using google maps geolocation

Users will want to search for providers who are local to them. One solution I considered was to ask providers to list a number of large towns / local areas that they serve. This probably would have worked OK but relies on both provider and client having the same idea of what constitutes "near", and also relies on places being spelled correctly. A neater solution was to utilise Google's maps platform (Google, n.d.).

In order to keep the form looking as clean as possible I started with a check-box to show or hide the location bits of the form. The code initialises a map using my Google maps API key and the callback function "load":

```
<script
src="https://maps.googleapis.com/maps/api/js?key=MyAPIKey&callback=load">
</script>
```

The load function sets some default options for the map location and zoom level, or, if we already have coordinates for the practice, uses them instead. It writes the map into the "mapDiv" using these "mapOptions":

```
var map = new google.maps.Map(mapDiv, mapOptions);
```

The map has a box that invites the applicant to enter their address or postcode. This is used to "geocode" their location: to obtain latitude and longitude coordinates and also to plot it on the map. This is achieved by using Google's geocode API (Google, n.d.):

```
var geocoder = new google.maps.Geocoder();
```

When the applicant enters their address into the address input field and clicks the "find location" button it calls the geocoder's geocode function using the value of the address input field as the input. If the geocoding is successful ("OK") then the call-back function sets the new centre and zoom level of the map, adds a marker, and updates the value of the two hidden input fields for latitude and longitude. It is the values of these hidden input fields that the form finally submits.

```
geocoder.geocode(
  {
    address: address,
  },
  function (results, status) {
    if (status === 'OK') {
      var location = results[0].geometry.location;
      map.setCenter(location);
      map.setZoom(14);
      new google.maps.Marker({
        map: map,
        position: location,
      });
      document.getElementById('1.latitude').value =
location.lat();
      document.getElementById('1.longitude').value =
location.lng();
    } else {
      alert('Geocode was not successful for the following
reason: ' + status);
    }
  }
);
```

Profile picture uploading

It was my original intention to invite providers to upload a profile picture to be used as a thumbnail for all their videos. In order to have a consistent look I wanted to be able to control the dimensions of this picture, and also to have the facility for the provider to crop it before it was saved. To achieve this, I explored using a plug-in called "Croppie" (Croppie, n.d.). I grappled with this plug-in for some time but could only get it to partially work so I abandoned my profile photo idea in favour of using thumbnails captured from the videos themselves. On reflection I prefer this solution as it gives the user a more honest idea of what a video is likely to contain.

Record videos intro

In addition to giving the provider information this page allows them to choose the categories of questions that they wish to answer. These will be passed to "videos-record.php" using GET variables.

Record videos

The video recording process is the most complex part of the site. Every time the main page is refreshed it pulls a single question at random from the database. The page is then controlled by a dedicated JavaScript file "recorder.js" which dynamically changes the screen elements the provider sees in each phase of recording. When the provider wishes to save a particular video and thumbnail the JavaScript calls "save-video-file-inc.php" which has the code that pushes it to a bucket in Google Cloud Platform.

Accessing the camera and microphone

The first thing the JavaScript file does is to access the provider's microphone and camera using the MediaCapture and Streams API (Griffith, 2018):

```
const constraints = { video: true, audio: true };
const mediaStreamObj = await
navigator.mediaDevices.getUserMedia(constraints);
const vidPreview = document.querySelector('video#vidLive');
vidPreview.srcObject = mediaStreamObj;
```

Before recording

Every time the "videos-record.php" page is refreshed it fetches a random question to ask the provider. This question needs to be one from the appropriate channel, that they haven't hidden, and they haven't already answered.

The SQL statement to do this is somewhat complex as we need to that the question does NOT EXIST in the "hidden_questions" table:

```
$sql = "SELECT * FROM questions WHERE questions.fk_channel = "
. $id_channel . " AND NOT EXISTS (
    SELECT fk_question FROM hidden_questions WHERE
questions.id_question = hidden_questions.fk_question AND
fk_user = " . $_SESSION['id_user'] .
") AND NOT EXISTS (
    SELECT fk_question FROM answers WHERE questions.id_question
= answers.fk_question AND fk_user = " . $_SESSION['id_user'] .
")
ORDER BY RAND() LIMIT 1";
```

When the question has been fetched it is displayed to the provider. They only see buttons relevant to choosing whether to answer that question as the JavaScript file sets the other screen elements so they are not displayed, for example:

```
btnRecNextQ.style.display = "inline";
btnRecStop.style.display = "none";
```

If the provider doesn't like the question and presses the "Hide" button the JavaScript sends an instruction to "save-hidden-question-inc.php" using an XMLHttpRequest (W3 Schools, n.d.):

```
var xhr = new XMLHttpRequest();
xhr.onreadystatechange = function () {
  if (this.readyState == 4 && this.status == 200) {
    location.reload(true);
  }
};
xhr.open('POST', 'includes/save-hidden-question-inc.php');
xhr.send();
```

The "save-hidden-question-inc.php" file saves the question id to the "hidden_questions" table in the database so they will never be asked it again. The "Answer later" button simply reloads the page. If the provider does not press a button the JavaScript runs a countdown timer before calling the "startRecording()" function.

During recording

The JavaScript creates a "mediaRecorder" object and an array to hold the data captured from the camera when data becomes available (Dutton, 2016):

```
let mediaRecorder = new MediaRecorder(mediaStreamObj);
let chunks = [];
mediaRecorder.ondataavailable = function (ev) {
  chunks.push(ev.data);
};
mediaRecorder.start();
```

During recording the JavaScript updates the screen with a count-up and flashes the border when there are just a few seconds to go. When the provider decides they are done (or runs out of time) the code stops the recording.

After recording

When the video recorder stops the code takes the chunks of video information from the array, and puts it into a binary large object (BLOB). Then it updates the source of the <video> tag to point to this blob so the provider can review their recording:

```
mediaRecorder.onstop = (ev) => {
  let blob = new Blob(chunks, { type: 'video/mp4;' });
  chunks = [];
  let videoURL = window.URL.createObjectURL(blob);
  vidReview.src = videoURL;
```

If the provider likes the video and chooses to save it then the code captures a thumbnail from the video (West, 2017) then sends it, along with the blob, to the "save-video-file-inc.php" page using another XMLHttpRequest.

The "save-video-file-inc.php" page performs two functions: it saves the files, and updates the "answers" table on the database.

Local storage

In my first implementation I saved the files locally. To save the video I needed to extract it from the global "\$_FILES" variable, extract it from the automatically created key called "tmp_name", then save it with the appropriate filename:

```
$vid_file = $_FILES['videofile'];
$vid_blob = file_get_contents($vid_file['tmp_name']);
file_put_contents($vid_file_path, $vid_blob);
```

Saving the thumbnail had its own complexity. I needed to explode the comma delimited file into an array, decode the second element of that array, then save it with the appropriate filename (Newdevzone, 2022):

```
$thumb_file = $_POST['thumbnail'];
$thumb_file_array = explode(',', $thumb_file);
$thumb_string = base64_decode($thumb_file_array[1]);
file_put_contents($thumb_file_path, $thumb_string);
```

Cloud storage

My second implementation was to save the files to Google Cloud Platform. This proved to be rather challenging to set up as I needed to download PHP to my computer, update the version of PHP that I was running in MAMP, and update the version of Composer that I was using before I could even download the google/cloud-storage component (the very first line in the instructions) (Google Cloud Client Library, n.d.).

To access cloud storage, I needed to set up and download a JSON access token which I then proceeded to share with the world when I uploaded it to my public GitHub repository(!). After cancelling this token, I was careful to save the next one to a folder that was ignored by Git.

In the code I first needed to link to the access token (learnWebCoding, 2020), autoload the libraries installed by composer, then import the Google Cloud client library:

```
putenv("GOOGLE_APPLICATION_CREDENTIALS=/Applications/MAMP/htdocs/ViewChoose/love/viewchoose-7d1ca3d99c67.json");
require __DIR__ . '/../vendor/autoload.php';
use Google\Cloud\Storage\StorageClient;
```

I created a bucket called "viewchoose" and copied the "upload_object()" function from the documentation (Google Cloud, n.d.). Once I had all this working, sending the files to the cloud was actually easier than saving them locally:

```

upload_object($bucket_name, $vid_filename,
$vid_file['tmp_name']);
upload_object($bucket_name, $thumb_filename,
$_POST['thumbnail']);

```

viewchoose

Public to Internet: This bucket is publicly accessible because allUsers or allAuthenticatedUsers have one or more permissions. Remove these principals to stop public access.

[EDIT ACCESS](#) [DISMISS](#)

Location
us (multiple regions in United States)

Storage class
Standard

Public access
Public to Internet

Protection
None

[OBJECTS](#) [CONFIGURATION](#) [PERMISSION](#) [PROTECTION](#) [LIFECYCLE](#)

Buckets > viewchoose

[UPLOAD FILES](#) [UPLOAD FOLDER](#) [CREATE FOLDER](#) [TRANSFER DATA](#) [MANAGE HOLDS](#) [DOWNLOAD](#) [DELETE](#)

Filter by name prefix only [Filter](#) Filter objects and folders [Show deleted data](#)

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified	Public access	Copy URL	Version history	
<input type="checkbox"/>	29-14.mp4	227.8 KB	video/mp4	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download
<input type="checkbox"/>	29-14.png	335.9 KB	image/png	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download
<input type="checkbox"/>	29-17.png	179.9 KB	image/png	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download
<input type="checkbox"/>	29-17.mp4	339.2 KB	video/mp4	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download
<input type="checkbox"/>	29-18.png	260.6 KB	image/png	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download
<input type="checkbox"/>	29-18.mp4	346.2 KB	video/mp4	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download
<input type="checkbox"/>	29-18.png	346.2 KB	image/png	2 Sept 20...	Standard	2 Sept 202...	Public to Internet	Copy URL	—	Download

Manage videos

In the event that a provider wants to delete a video this page will call an included PHP file that deletes the video and thumbnail from the cloud (Google Cloud, n.d.) and also from the database.

Testing

In the course of the project, I performed three types of testing: unit testing, integration testing, and user testing.

Unit testing

When creating a new piece of code my strategy is always to start off as simple as possible and then to add complexity. For example, if a button is to call a particular function, I will start by having the function create and alert box so that I can be sure that the event listener is working on the button before I write anything else. In addition to alert boxes I make much use of "console.log()" functions to print the values of variables to the console. When programming in PHP I will often echo the values of variables onto the screen.

Where possible I will feed a function with test cases so I can check the outputs. When I can't access the inputs of the function directly I will recreate scenarios in the user interface and then check that the function is doing what I'm expecting, for example, that a function to save data to the database is actually doing just that.

For new areas of learning I will start by creating a sandbox page that is as close as possible to the examples given in the online documentation. Once I've got the sandbox to work, I have one foot on the ground and I can start integrating the functionality into the main code. A good example of this was when I was learning how to save to the cloud: I followed the documentation and a YouTube video until I finally got a simple example to work. When I transferred that functionality into my own code I was baffled when it stopped everything else from working. I was able to delete line by line until I found the problem line and (eventually) realised that I was now running the code from within a folder and so this needed to be reflected in the path I was using to access the library.

The most difficult aspect of unit testing was trying to debug PHP code in an included PHP file, for example when "player.js" sends changes to a user's favourites to "save-fav-poster-inc.php" for saving to the database. I couldn't find a good way to monitor what was happening inside the PHP file: I couldn't work out how to output variable values to the console, echo them to a page (as the included file is not shown on screen), nor return them to the calling page. To work around this, I created a session variable and put my variable values and error messages into that. I echoed that session variable out on the page that called the function and, when I refreshed the page, was able to see what was happening within the included function. I hope and assume there is an easier way that I haven't yet found.

Integration & user testing

In addition to talking to many people casually about the project I conducted 5 formal interviews with potential users of the site. In order to test the site from end to end I invited each interviewee to interact with the site as if they were looking for a therapist and then as if they were a provider who wanted to post videos to the site.

I was aware that there was a risk that interviewees would tell me what they thought I wanted to hear. To counteract that I invited each interviewee to be as critical as possible, thanked them whenever an error arose, provided as little guidance as possible during the testing process, and used "clean" questioning techniques (Wikipedia, n.d.) to get them to explore and expand upon their perceptions.

This end-to-end testing proved to be very useful as I was able to identify many technical issues as we worked through the various pages of the site. I was also able to gain multiple perspectives on the branding and general concept of the site which have been invaluable in deciding how to take the project forward.

My notes from these meetings are in Appendix 1. The following table shows the actions taken as a result:

Interviewee	Action	Status
JL	Fix the screen when the poster has completed all the questions.	DONE
JL	Add "Languages spoken" to provider profile.	
DT	Remove the menu bar from the index page to make it less confusing and to ensure users choose a channel.	DONE
DT	Make index page banner smaller so it is less overwhelming.	DONE
DT	Re-phrase the "become a poster" menu item to "post videos" and take people to the user sign-up page if they are not yet signed in.	DONE
DT	Make the main video smaller so that the buttons are visible.	DONE
VP	Experiment with tabs instead of having an index page for choosing channel.	DONE
VP	Put pauses between videos.	DONE
VP	If there are no videos in the search ensure that there are no ugly video placeholders in the queue.	DONE
VP	Add tooltips to the video playback buttons.	DONE
VP	Change "answer questions" to "record videos".	DONE
VP	Make preview video when recording smaller.	DONE
VP	Add total likes to the videos when watching.	
VP	Fix the issue that caused "Tommy" not to show.	DONE
VP	Fix the email contact so that it sends to the poster email and not to me.	DONE
TP	Fix unsigned-in profile button issue.	DONE
TP & AS	Stop auto-playing videos when you first access the page.	
TP	Add tabs to the top of the page for channels.	DONE
AS	Change user interface to reference "Providers".	DONE
AS	Fix sign-in page login error.	DONE
AS	Change categories to "Show me videos about ..."	DONE
AS	Change position and words on the profile button.	
AS	Include profile information to the right of playing video.	

Interviewee	Action	Status
AS	Include a "Welcome back" message with the logged-in person's name in the menu bar.	DONE
AS	Add profile picture to top of profile page.	DONE
AS	Make map smaller on profile page.	DONE
AS	Fix the gap under the map.	DONE
AS	Re-word profile contact form.	DONE
AS	Make contact form disappear once message is sent.	DONE

In addition to the improvement actions identified in the user testing interviews I have identified the following further improvements:

Page / file	Possible improvement
Navigation	Include an option for managing the user-profile. Include password recovery.
Browse videos	Refine keyword search to associate words, e.g. Anxious and Anxiety. Add more filters. Get clear on nomenclature: am I working with likes / dislikes or favourites? There is currently a lot of repeated code to save to the two favourite tables. Explore the possibility of re-factoring into a single function.
Browse profiles	Add code snippet functionality.
Ask a question	Add validation to the input fields.
User sign-up	Create sign-up confirmation email. Refactor so it uses session variables rather than GET variables for error messages and persisting data. Resolve issue of mental health navigation bar appearing when signing up from the welcome page. Gather location information for use in proximity searches. Enable the same page to be used for updating user information.
Favourites	Expand to allow the user to manage providers and videos that have been disliked. Access shared functions for updating database.
Provider sign-up	Enable the same page to be used for updating user information. Make multi-channel.

Record videos intro	Send category information to the "Record videos" page.
Record videos	<p>Add error handling for accessing the camera and microphone.</p> <p>Record videos in a format which is readable in Safari and Firefox browsers.</p> <p>Filter questions according to desired categories.</p>

Reflections

As I had already built a prototype for my project proposal, I had a clear idea of what I wanted to achieve and I felt well prepared to start this project.

Thinking I could learn a new technology stack was clearly over-optimistic but I was able to identify this early enough to change tack and start getting some useful work done. That said, I'm still curious about building this in a MERN stack and will likely look into this after this report has been submitted.

The technical side was predictably challenging - especially the video recording and saving. Accessing data and moving it from PHP to JavaScript, then back to PHP was rather complex and clunky. Implementing a relatively simple feature like favourites proved to be much more complex to achieve than I expected. On the flip side, formatting using the Bootstrap library proved to be relatively simple and I was able to write some rather complex queries in SQL that did what I needed them to do.

In addition to learning and practicing my coding skills a really useful aspect of the project has been to be able to present potential users with an advanced working prototype. People who I have spoken to before, and who were enthusiastic about the concept of the site, were able to give much more specific feedback about what they liked and disliked.

A key takeaway for me is that leading out with a plethora of randomised questions and answers - especially when they auto-play - is overwhelming for users. Rather than creating a video Q&A site with the added bonus that you can contact a provider I now think I should be creating a more focussed "find-a-provider" site backed up by a video Q&A.

In my new vision for the site each provider must maintain one (or a few) "intro" videos in order to remain listed. Though these will be recorded through the site they will not have the spontaneous nature of the other videos as providers will be able to practice them again and again. In addition to these core videos they will have the option of creating more spontaneous video content that answers frequently asked questions. Rather than the main page of each channel being a video answer browser, it will be a provider browser consisting solely of introductory videos. If a single video is enough for the user to feel that they vibe with the provider then I will make it easy for them to initiate contact. If they do want to see more detail then they will have the option to dig deeper.

With this sharpening of focus in mind I will now be asking people for their thoughts on how "ViewChoose.com" compares to a new alternative brand: "MeetTheProvider.com":



View Choose



Meet The Provider

Bibliography

- Agarwal, N., 2020. *PHP: Get Values of Multiple Checked Checkboxes*. [Online]
Available at: <https://www.formget.com/php-checkbox/>
[Accessed 10 07 2022].
- Bootstrap, n.d. *Build fast, responsive sites with Bootstrap*. [Online]
Available at: <https://getbootstrap.com>
[Accessed 09 09 2022].
- Bootstrap, n.d. *Collapse*. [Online]
Available at: <https://getbootstrap.com/docs/5.2/components/collapse/>
[Accessed 15 09 2022].
- Bootstrap, n.d. *Grid system*. [Online]
Available at: <https://getbootstrap.com/docs/5.2/layout/grid/>
[Accessed 15 09 2022].
- Bootstrap, n.d. *Navs and tabs*. [Online]
Available at: <https://getbootstrap.com/docs/5.2/components/navs-tabs/#tabs>
[Accessed 11 09 2022].
- brapifra, n.d. *PHP Server*. [Online]
Available at: <https://github.com/brapifra/vscode-phpserver>
[Accessed 02 07 2022].
- CodexWorld, 2017. *How to Convert PHP Array to JavaScript Array*. [Online]
Available at: <https://www.codexworld.com/how-to/convert-php-array-to-javascript-array/>
[Accessed 09 08 2022].
- com, z. a. z. d., n.d. *session_unset*. [Online]
Available at: <https://www.php.net/manual/en/function.session-unset.php>
[Accessed 11 07 2022].
- Croppie, n.d. *Croppie*. [Online]
Available at: <http://foliotek.github.io/Croppie/>
[Accessed 15 07 2022].
- Dey, R., n.d. *Live Server*. [Online]
Available at: <https://github.com/ritwickdey/vscode-live-server>
[Accessed 02 07 2022].
- Dey, R., n.d. *Live Server Web Extension*. [Online]
Available at: <https://chrome.google.com/webstore/detail/live-server-web-extension/fiegdmejfeffgpnejdinekhficiaogmj/>
[Accessed 02 07 2022].
- Dutton, S., 2016. *Record audio and video with MediaRecorder*. [Online]
Available at: <https://developer.chrome.com/blog/mediarecorder/>
[Accessed 12 09 2022].
- Evans-Lacey, R., 2022. *MSc Project Proposal*. s.l.:Birkbeck.
- Font Awesome, n.d. [Online]
Available at: <https://fontawesome.com>
[Accessed 15 09 2022].

Google Cloud Client Library, n.d. *Google Cloud Storage for PHP*. [Online]
Available at: <http://googleapis.github.io/google-cloud-php/#/docs/cloud-storage/v1.28.1/storage/readme>
[Accessed 12 09 2022].

Google Cloud, 2022. *Make data public*. [Online]
Available at: <https://cloud.google.com/storage/docs/access-control/making-data-public#prereq-console>
[Accessed 01 09 2022].

Google Cloud, n.d. *Delete objects*. [Online]
Available at: <https://cloud.google.com/storage/docs/deleting-objects#storage-delete-object-php>
[Accessed 12 09 2022].

Google Cloud, n.d. *Upload objects*. [Online]
Available at: <https://cloud.google.com/storage/docs/uploading-objects#prereq-code-samples>
[Accessed 12 09 2022].

Google, n.d. *Geocoding API*. [Online]
Available at: <https://developers.google.com/maps/documentation/geocoding/requests-geocoding>
[Accessed 15 07 2022].

Google, n.d. *Google Maps Platform*. [Online]
Available at: <https://developers.google.com/maps>
[Accessed 15 07 2022].

Griffith, S., 2018. *Capturing and Saving User Audio or Video with JavaScript*. [Online]
Available at: <https://www.youtube.com/watch?v=K6L38xk2rkk>
[Accessed 12 09 2022].

kn000x, 2014. *how to display errors on MAMP?*. [Online]
Available at: <https://stackoverflow.com/questions/21784871/how-to-display-errors-on-mamp>
[Accessed 02 07 2022].

Krossing, D., 2018. *How To Create A Login System In PHP For Beginners | PHP Tutorial*. [Online]
Available at: <https://youtu.be/LC9GaXkdxF8>
[Accessed 11 09 2022].

learnWebCoding, 2020. *GCP Part6: Setup Google Cloud Storage PHP Library*. [Online]
Available at: https://www.youtube.com/watch?v=6tdNoWqsRqk&ab_channel=learnWebCoding
[Accessed 12 09 2022].

Logos-World, 2017. *YouTube Logo*. [Online]
Available at: <https://logos-world.net/youtube-logo/>
[Accessed 08 09 2022].

MAMP, n.d. *MAMP - The free web development solution with Apache, Nginx, PHP & MySQL*. [Online]
Available at: <https://www.mamp.info/en/mamp/mac/>
[Accessed 02 07 2022].

Mewburn, B., n.d. *Intelephense*. [Online]
 Available at: <https://github.com/bmewburn/vscode-intelephense>
 [Accessed 02 07 2022].

Newdevzone, 2022. *How to save a PNG image server-side, from a base64 data URI*. [Online]
 Available at: <https://newdevzone.com/posts/how-to-save-a-png-image-server-side-from-a-base64-data-uri>
 [Accessed 12 09 2022].

php, n.d. *mail*. [Online]
 Available at: <https://www.php.net/manual/en/function.mail.php>
 [Accessed 11 09 2022].

php, n.d. *password_verify*. [Online]
 Available at: <https://www.php.net/manual/en/function.password-verify.php>
 [Accessed 11 09 2022].

Smilga, J., 2020. *Full React Course 2020 - Learn Fundamentals, Hooks, Context API, React Router, Custom Hooks*. [Online]
 Available at: <https://www.youtube.com/watch?v=4UZrsTqkcW4&t=26348s>
 [Accessed 21 06 2022].

Smilga, J., 2021. *Node.js / Express Course - Build 4 Projects*. [Online]
 Available at:
https://www.youtube.com/watch?v=qwfE7fSVaZM&ab_channel=freeCodeCamp.org
 [Accessed 30 06 2022].

Smilga, J., 2021. *Node.js and Express.js - Full Course*. [Online]
 Available at:
https://www.youtube.com/watch?v=Oe421EPjeBE&t=24922s&ab_channel=freeCodeCamp.org
 [Accessed 21 06 2022].

Stack Overflow, n.d. [Online]
 Available at: <https://stackoverflow.com/questions/5098397/how-to-pass-an-array-in-get-in-php>
 [Accessed 06 07 2022].

The Medium Well, n.d. *Responsive vs Adaptive Design – Which is Best for Mobile Viewing of Your Website?*. [Online]
 Available at: <https://mediumwell.com/responsive-adaptive-mobile/>
 [Accessed 09 09 2022].

Traversy, B., 2022. *PHP For Beginners | 3+ Hour Crash Course*. [Online]
 Available at: <https://www.youtube.com/watch?v=BUCiSSyIGGU>
 [Accessed 01 07 2022].

Visual Studio Code, n.d. *Visual Studio Code*. [Online]
 Available at: <https://code.visualstudio.com>
 [Accessed 02 07 2022].

W3 Schools, n.d. *PHP MySQL Prepared Statements*. [Online]
 Available at: https://www.w3schools.com/php/php_mysql_prepared_statements.asp
 [Accessed 12 09 2022].

W3 Schools, n.d. *XML HttpRequest*. [Online]
Available at: https://www.w3schools.com/xml/xml_http.asp
[Accessed 12 09 2022].

Web Accessibility Initiative, 2019. *Labeling Controls*. [Online]
Available at: <https://www.w3.org/WAI/tutorials/forms/labels/>
[Accessed 16 07 2022].

West, C., 2017. *Get Video Frame As An Image*. [Online]
Available at: <https://cwestblog.com/2017/05/03/javascript-snippet-get-video-frame-as-an-image/>
[Accessed 12 09 2022].

Wikipedia, n.d. *Clean Language*. [Online]
Available at: https://en.wikipedia.org/wiki/Clean_language
[Accessed 13 09 2022].

Wikipedia, n.d. *SQL Injection*. [Online]
Available at: https://en.wikipedia.org/wiki/SQL_injection
[Accessed 12 09 2022].

Appendices

Appendix 1: User acceptance testing meeting notes

JL, potential user, 7 August 2022

Has had lots of experience looking for therapists and thought the videos would really help her get to know someone.

Didn't really vibe with the name "ViewChoose" but couldn't think of anything that she would prefer.

Didn't like the name "Posters" but couldn't think of anything better that would apply to all channels.

Optional location for user sign-up so we can use proximity searching.

Could include "Languages spoken" in poster profile.

How to handle USA based therapists whose licences prevent them from practicing outside of their state?

All the buttons showed once all the questions were answered.

DT, psychotherapist, 26 August 2022

Liked the "YouChoose" logo on the index page but thought the banner was rather large.

Thought the use of the word "vibe" was rather casual but fitted with the informal theme.

Thought the buttons and explanation were too small in relation to the banner.

Thought "Like the posters who speak to you most" should be re-worded to "Like your favourites".

Thought the "Become a poster" button was too buried as he couldn't find it on the index page.

Thought the process of signing up as a user then becoming a poster could be more clearly signposted.

When trying to sign up put his email and a password into the boxes intended for logging in.

The main video on the watch screen was so large that the buttons underneath were not visible.

The videos in the list were not clickable so you couldn't jump to one later in the list that looks interesting.

Liked that when recording it wasn't a mirror image but rather how someone would see you in real life.

Thought therapists might be reluctant to sit down and create videos.

VP, potential client, 28 August 2022

Wondered if, rather than having an index page, there could be tabs for each channel across the top of the screen?

Found the transition between videos when they were playing somewhat jarring and suggested a pause between them. Perhaps giving the viewer the option to watch another one rather than autoplaying from the start.

When the keyword "thing" was used no videos were retrieved and the places where the videos would have been had ugly placeholders.

Was not clear from the icons what the like and profile buttons do. Is it possible to have a description when the mouse hovers over them? Maybe have a button like: "Go to Vlad's profile" at the top or the other buttons further over to the sides of the screen?

Was not clear on the sign-up page if all fields are "required" and they could be labelled as such.

Would like to see how many "likes" each video has.

Would prefer the menu item "answer questions" to say "record videos".

Found the video preview when recording videos is too big.

Thought it would be nicer to offer people a choice of three thumbnails (from the beginning, middle, and end) rather than having them choose one by reviewing the video.

When recording thought it would be nice to have a number showing how many questions are left to be answered.

When all videos have been answered the screen got messy - including a "review" button that does not get used.

When selecting a "Tommy" thumbnail from the index screen no answers were found.

Profile mail not sending to the correct address - instead it always goes to "richard@psychicplumbing.com".

Poster email did not change when database was updated as it is stored in a session variable.

Maybe it is too confusing to let users have different emails for their user profile and their poster profile?

TP, Mindfulness Coach, 2 September 2022

Liked the initial YouChoose banner in red and white and said they were the best colours for marketing.

Was initially against the wording of finding someone to "vibe with" but later decided she liked it.

Wondered why the site was centred around questions rather than allowing the user to search for the type of practitioner they are looking for who is local to them. Thought the answers to questions told her nothing about who they are but did like that the profiles had questions at the bottom.

Would it be possible to have a more traditional list of profiles and, maybe, filter for them in the same way as for videos?

Noticed when not signed in that clicking the profile button under the video takes you to the profile but that the profile then had error messages on it.

Would definitely like to filter by speciality. Noticed that "Mindfulness Coach" was not on the list of specialities.

Found the immediate recording was off-putting and would prefer some kind of intermediate page with tips on recording a good video.

Thought that tabs across the top would be good.

AS - Meditation teacher, 7 September 2022

Thought the ViewChoose logo looked spooky and haunted. Thought the brand name sounded like a video site rather than somewhere to find a professional. Would have preferred to see the faces of friendly practitioners and a warmer colour pallet.

Thought "posters" sounded like printed paper and preferred the word "providers".

Liked the progression of steps on the welcome page but thought it would be clearer as a process if there were arrows in-between each icon. Thought this process could be more clearly signposted throughout the site.

Noticed that there was a "header info" error when signing in on the sign-up page. Thought the sign-up page should ask for location information so the site only shows providers close to you.

Didn't like that the videos auto played - at least there should be some kind of count down. Thought that it would be good to display some basic profile information with the video as it was playing and to include calls to action after every video: "Book now", "Watch more videos about ...", "Watch more videos from ...".

Thought the keyword field should link words like "anxious" with "anxiety".

Thought the categories of questions were confusing. Would be better if it was phrased "Show me videos about..."

Didn't like having the provider's name on the question, wants the site to talk to him.

Found it difficult to distinguish between the formatting for a liked and unliked video.

Thought there should be a profile picture at the top of the profile page. Would prefer if the contact form didn't like the phrase "work together" as it wouldn't be appropriate for other channels. Thought the map should be smaller and hours of availability should be included. Thought the gap under the map was too small. Thought a number of smaller boxes would be less intimidating than a single box for the contact form. Once contact has been made thought it would be nice to play a short video or at least move the "sent" message to the top.

Thought there should be 3 types of provision: online, you come to me (with a point location), and I come to you (with an area of operation).

Questioned the rationale of providing a repository of video answers with the added bonus of being able to contact a provider - thought YouTube already did that. Instead, thought it would be better to be closer to a standard "find a" website but with videos. Have a core set of between 4 and 7 introductory videos that were necessary for each video profile and ask the providers to contribute to a wider Q&A as the added bonus.