



User Manual

7th May 2021

Students:

Janesh Sharma - 17473124

Paul McNally - 17318221

Supervisor:

Dr. Stephen Blott

Table Of Contents

Table Of Contents	2
1. Overview	3
2. Installation	3
2.1 Prerequisites	3
2.2 Setup	4
2.2.1 Client	4
2.2.2 Backend	4
2.2.3 Remote Code Execution	4
3. How To Run	4
3.1 Prerequisites	4
3.2 Run Locally	4
3.2.1 Client	4
3.2.2 Backend	5
3.2.3 Remote Code Execution	5
4. Introduction	5
4.1 Registration	5
4.2 Login	7
4.3 Dashboard	7
4.4 Delete Account	8
4.5 Navbar	9
5. Create Coding Test	10
5.2 Create Test	10
5.3 Setup Challenges	10
5.4 Setup Questions	12
6. Edit	13
6.1 Edit Test	13
6.2 Edit Challenges	14
6.3 Add Challenges	14
6.4 Edit Questions	15
6.5 Add Questions	15
6.6 Add Participants	16
6.7 Reset Test	16
7. Results	17
7.1 Participants	17
7.2 Individual Results	18
7.3 Challenge Results	18
7.4 Video Responses	19
8. Coding Test	19

1. Overview

Coding Test Platform is a full-stack web app for a coding test platform allowing companies to assess an applicant's coding ability.

It is useful for any software related business that needs to hire and assess software developers' applications. Our system allows businesses to effectively assess an applicant's coding skills by providing a coding test platform, video responses to employer questions and analytics to rank the best candidates.

This projects features include:

- Microservices Architecture
- Authentication
- IDE
- Remote Code execution with Docker
- Video responses
- Analytics
- Email

This Project was built using a React frontend, Node.js backend with Jest for testing, MongoDB as a database and a full CI/CD workflow on GitLab which deploys to Google's Firebase hosting and serverless cloud functions.

2. Installation

If you don't want to run this project locally you can visit the production version at [Coding Test Platform](#).

Note due to the use of Firebase, it has security measures in place which means a user must be added to the project and sign in on the command line via the command `firebase login` before gaining access to the project keys and configs necessary to run parts of the project. Thus, 'how to run' will not work for you unless you are added to the project on firebase console. Please visit the hosted version link above.

2.1 Prerequisites

1. Install [Node](#) LTS (version 14.16.1 at the time of writing).
2. Install [Docker](#).
3. Install [Git](#).

2.2 Setup

Navigate to the inside of the project directory and enter the following commands:

2.2.1 Client

1. `cd src/client`
2. `yarn`

2.2.2 Backend

1. `cd src/functions`
2. `npm i`

2.2.3 Remote Code Execution

1. `cd src/remoteCodeExecuton`
2. `npm i`

3. How To Run

This project is hosted publicly and available at [Coding Test Platform](#).

Note due to the use of Firebase, it has security measures in place which means a user must be added to the project and sign in on the command line via the command `firebase login` before gaining access to the project keys and configs necessary to run parts of the project. Thus, 'how to run' will not work for you unless you are added to the project on firebase console. Please visit the hosted version link above.

3.1 Prerequisites

1. Install the Firebase CLI using npm via the following command:

```
npm install -g firebase-tools
```

3.2 Run Locally

Navigate to the inside of the project directory and enter the following commands:

3.2.1 Client

1. `cd src/client`
2. Replace the contents of `.env` with the following to ensure the local APIs are used:

```
REACT_APP_ENV=development
```

```
REACT_APP_ALT_RCE_API=false  
# REACT_APP_RCE_API=https://35.225.18.17
```

3. yarn start

3.2.2 Backend

1. `cd src/functions`
2. `firebase serve --only functions`

3.2.3 Remote Code Execution

1. Start Docker
2. Run the following commands to download the necessary images required for remote code execution to work.

```
docker pull redis:alpine  
docker pull python:3-alpine  
docker pull openjdk:8-alpine  
Docker pull node:lts
```

3. Start redis queue using docker:

```
docker run -d -p 6379:6379 --name queue redis:alpine
```

4. Run the server and workers

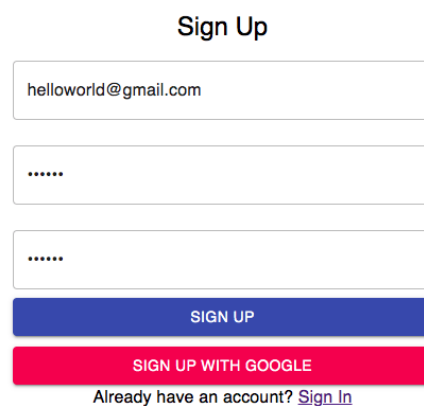
```
cd src/remoteCodeExecuton  
npm run server  
npm run worker
```

The Application is now accessible at <http://localhost:3000>.

4. Introduction

4.1 Registration

When a user first uses the application they will need to create an account in our database. They can either sign up with email and password or just sign up with google if they wish. To sign up with email and password they must enter a valid email and create a password that is at least characters long. To login with google, the user must already have an existing google account click “sign up with google” button as shown below in figure 1.1



The image shows a 'Sign Up' form with the following elements:

- Sign Up**: Title of the form.
- : Email input field.
- : Password input field.
- : Confirm password input field.
- SIGN UP**: A blue button to submit the form.
- SIGN UP WITH GOOGLE**: A red button for Google authentication.
- Already have an account? [Sign In](#): A link for existing users.

Figure 1.1

After the user successfully creates an account they are then directed to a screen (see figure 1.2) to enter the company they work and are hiring for. After this, the user will be brought to the employer dashboard.

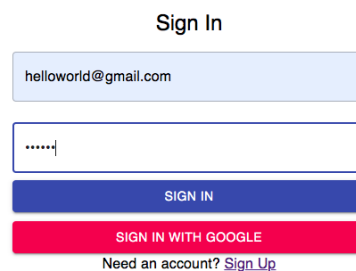
Enter Company

SUBMIT

Figure 1.2

4.2 Login

If the user already has an existing account, they can simply log in with their login credentials or else click the sign in google with google button. Once the user logs in they will be brought to the employer dashboard. If the user clicks sign in with google before they have signed up with google the application will handle this and bring them to the company input screen.



The image shows a 'Sign In' form. At the top, the text 'Sign In' is centered. Below it is a light blue rectangular input field containing the email address 'helloworld@gmail.com'. Underneath the email field is a white rectangular input field for a password, with the text '.....' indicating masked characters. Below the password field are two buttons: a blue button with the text 'SIGN IN' and a red button with the text 'SIGN IN WITH GOOGLE'. At the bottom of the form, centered, is the text 'Need an account? [Sign Up](#)'.

Figure 1.3

4.3 Dashboard

When a user successfully logs in they are brought to the dashboard, from there they can navigate to 3 different sections. If they click on the setup card or button they will be able to create a new coding test, set a number of coding challenges and also have the option of adding questions to be asked in the video interview.

By clicking on the edit existing coding test card or edit button the user will be able to see all of their existing previously created coding tests and have the option to perform various editing actions on them.

By clicking on the view previous tests history/results card or the results button they will be able to view the results of the coding tests taken by various participants.

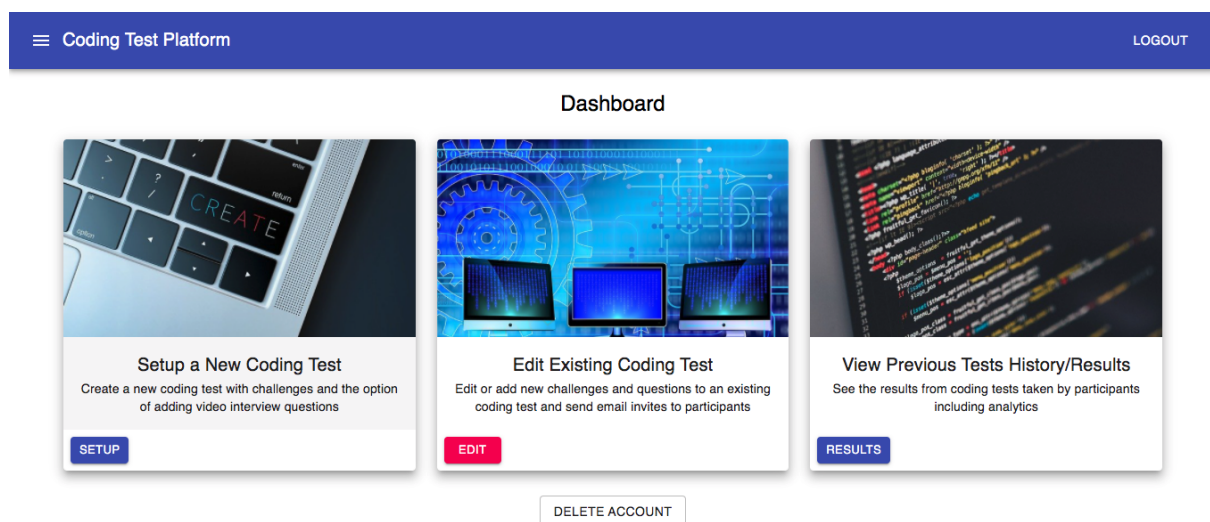


Figure 1.4

4.4 Delete Account

If a user chooses to delete their account, they must click the delete account button(see figure 1.5) on the dashboard and then confirm that they wish to delete their account. By deleting an account all data related to that user will be removed from our database this includes user credentials, user authentication, all coding tests they created and coding challenges they said up as well as all the results from the coding tests they emailed to participants. Any email invitations to coding tests not yet attempted by participants will be made invalid.

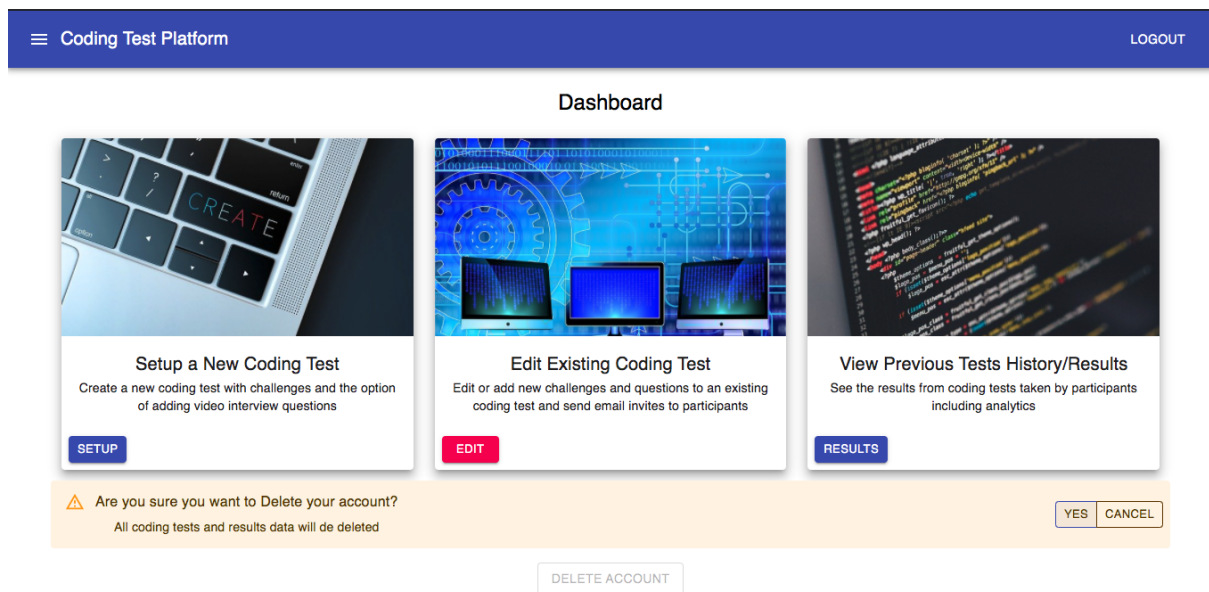


Figure 1.5

4.5 Navbar

The Navbar has a navigation drawer that can be used by the user anywhere in the employer ui, this makes navigating around the user interface easier and quicker from the navigation drawer the user can:

- Go to the Dashboard
- Go to setup a new coding test
- Go to edit existing coding test
- Go to view previous tests results page
- Logout

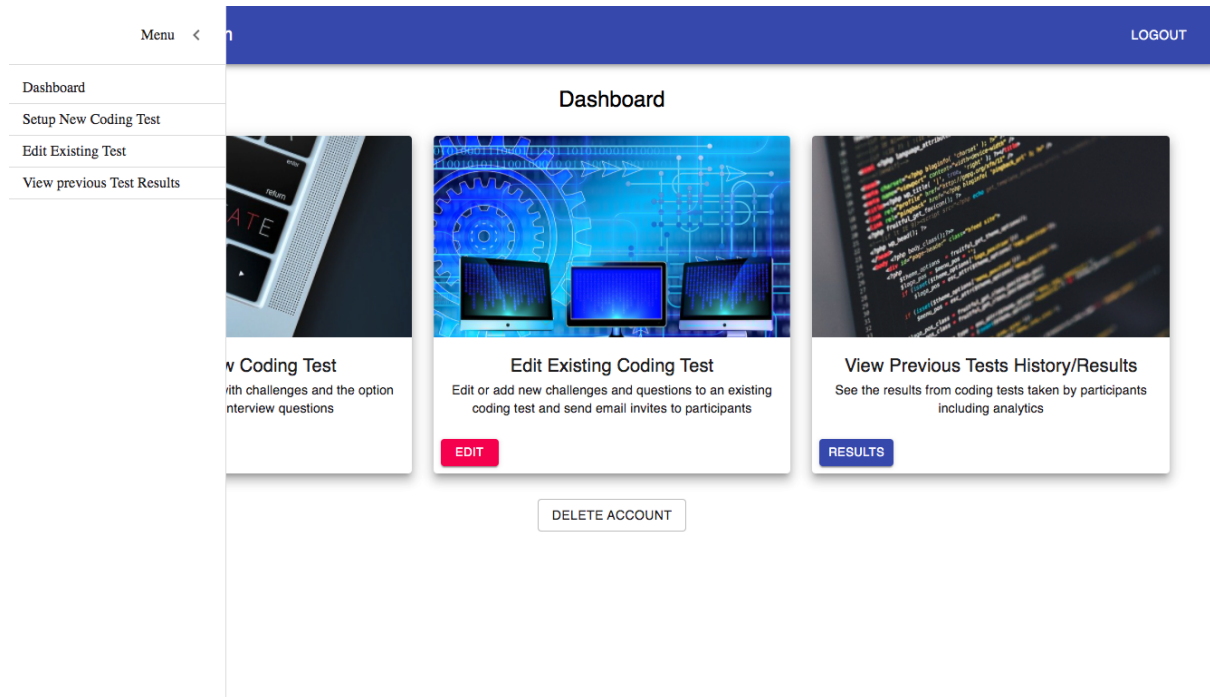
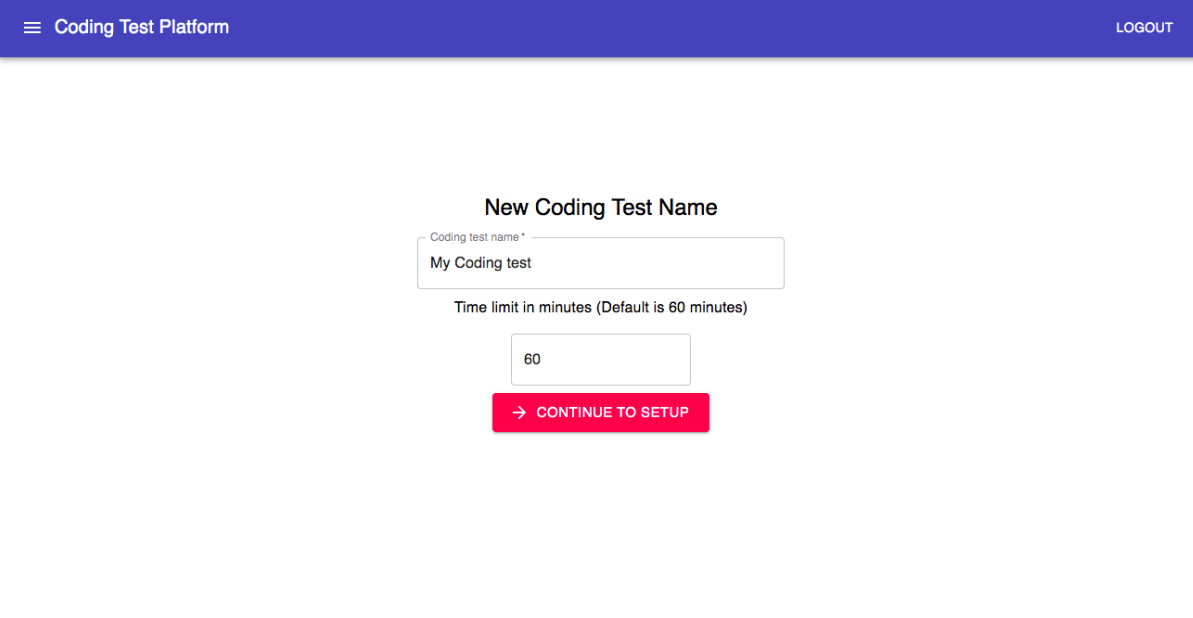


Figure 1.6

5. Create Coding Test

5.2 Create Test

When the user clicks on the setup card or button they will be brought to create a coding test screen. Here the user will have to enter a name for a new coding test. The user cannot have the same name for more than one coding test, therefore if they enter the name of an already existing test they will not be able to proceed to the setup unless they enter a new name or delete the existing test with that name. The user must also enter the time limit for the participants attempting the coding test. After they have entered a valid coding test name they will be able to continue to set up.



The screenshot shows a web interface for creating a new coding test. At the top, there is a dark blue header bar with a hamburger menu icon and the text 'Coding Test Platform' on the left, and a 'LOGOUT' link on the right. The main content area is white and contains a form titled 'New Coding Test Name'. The form has two input fields: the first is labeled 'Coding test name *' and contains the text 'My Coding test'; the second is labeled 'Time limit in minutes (Default is 60 minutes)' and contains the number '60'. Below these fields is a red button with a right-pointing arrow and the text 'CONTINUE TO SETUP'.

Figure 2.1

5.3 Setup Challenges

When the user gets to the setup challenge screen they will be required to fill out the required fields in the form. To assist them, there is an example problem provided (figure 2.2) which shows what information should be provided in the form.

The image shows a 'Coding Test Platform' interface. A modal window titled 'Example Coding Problem' is open, displaying the following information:

- Challenge Title:** Arrays: Left Rotation
- Problem Description:** A left rotation operation on an array shifts each of the array's elements unit to the left. For example, if left rotations are performed on array , then the array would become . Note that the lowest index item moves to the highest index in a rotation. This is called a circular array. Given an array of integers and a number, , perform left rotations on the array. Return the updated array to be printed as a single line of space-separated integers. Complete the function `rotLeft` in the editor below.
- Input Format:** The first line contains two space-separated integers n and d , the size of a and the number of left rotations. The second line contains n space-separated integers, each an $a[i]$.
- Return Format:** `int a[n]: the rotated array`
- Constraints:**
 - $1 \leq n \leq 10^5$
 - $1 \leq d \leq n$
 - $1 \leq a[i] \leq 10^6$

A 'CLOSE' button is located at the bottom right of the modal. The background shows a form for creating a challenge with fields like 'Title E.g. 'FizzBuzz'*', 'Problem Description of Challenge', 'Input Format', 'Return Format', and 'Constraints'.

Figure 2.2

The challenge setup form shown below (figure 2.3) has 8 required fields (Challenge title, problem description, input format, return format, constraints, sample input, sample output, example with explanation). The minimum and default time for the test cases is 15 seconds, users have the option to increase this if they wish. Users can provide their own test cases and are required to provide at least one in the form. When finished filling out the form the user clicks save challenge to save the challenge, they can add as many challenges as they wish. By clicking the continue button they will be brought to add questions screen, or else they can click exit to exit the setup test process.

Coding Test Platform LOGOUT

Setup A New Coding Test

[SEE EXAMPLE CHALLENGE](#)

Challenge Title
Title E.g. "FizzBuzz"
Reverse a string

Problem Description
Reverse a string from from `sys.argv` and print to stdout

Input Format
String

Return Format
String

Constraints
Input string less than 10000 characters

Sample Input
hello world!

Sample Output
`ldrow olleh`

Example with Explanation
hello world! reversed is `ldrow olleh`

Timeout for test cases (Minimum and default is 15 seconds)
15

Test Cases (Please provide at least 1 test case)

123	321
Input	Expected Output
dcu	ucd

[+](#) [-](#)

[SAVE CHALLENGE](#) [CONTINUE](#) [EXIT](#)

Figure 2.3

Everytime a user saves a challenge, an alert will be displayed at the top of the page to confirm to them that the challenge has been successfully saved in the database (figure 2.4).

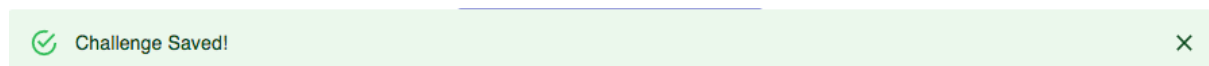


Figure 2.4

5.4 Setup Questions

After the user is finished adding coding challenges to the test, they then have the option to add up to 3 video interview questions. This is an optional feature so if the user does not wish to add questions they will have to click the skip and exit button and they will be brought back to the dashboard. Else they enter their questions, click the save and exit button and their questions will be saved in the database and the user will be brought back to the dashboard screen.

Coding Test Platform

LOGOUT

Video Interview Questions(optional)

Question 1

Question 1

Question 2

Question 2

Question 3

Question 3

SAVE AND EXIT

SKIP AND EXIT

Figure 2.5

6. Edit

After clicking on the edit existing coding test card or edit button on the dashboard the user will be able to see all of their existing previously created coding tests. Here the user can perform several actions, if they wish to add participants to a coding test and send out email invitations they can click the add participants button, to reset a test they click the reset test button, to edit the test the user clicks the edit icon. The user also has the option to delete the test by clicking the delete icon, by doing this all data related to the test will be removed from the database which includes challenges, questions, video responses and coding test results. Any email invitations for the deleted coding test not yet attempted by participants will be made invalid.

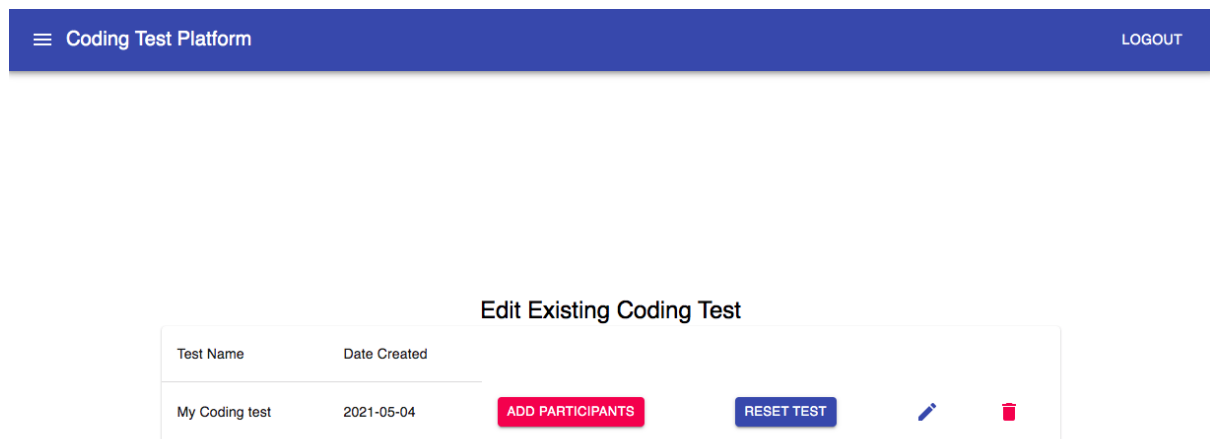


Figure 3.1

6.1 Edit Test

If the user clicks the edit icon they will be brought to the edit test screen, here they can see all current coding challenges and questions for that particular coding challenge. There is the option to edit or delete the current challenges and add new challenges to the test. There is also the option to edit or delete the current questions asked or add questions if none have been added previously. The user can also add participants to the coding test from this screen too.

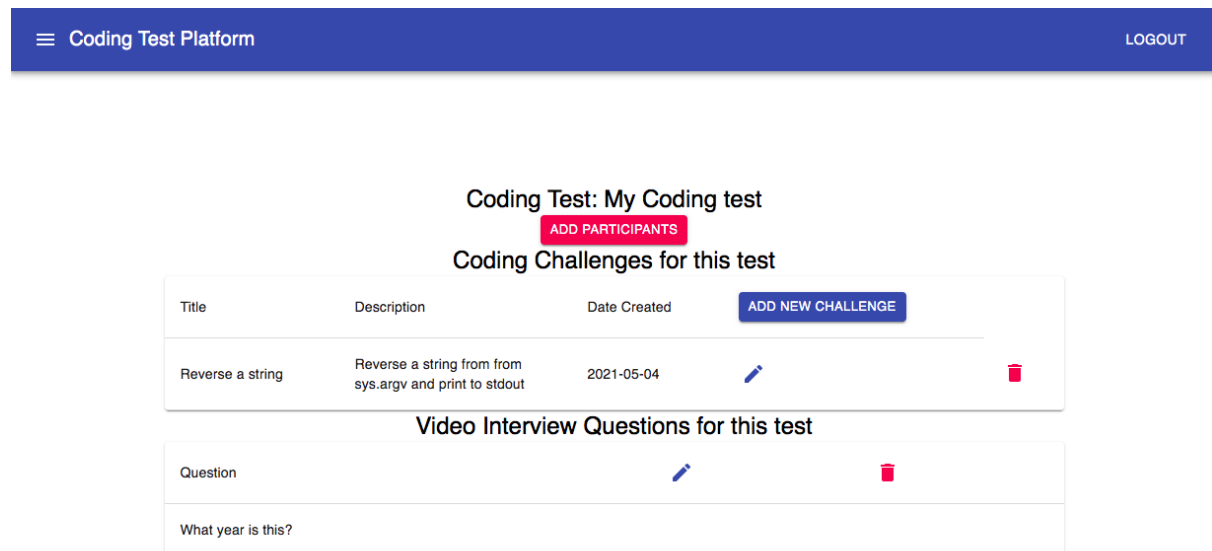


Figure 3.2

6.2 Edit Challenges

When a user clicks the edit icon for a challenge in the edit test screen they will be brought to the edit challenge form where all the information about the current challenge is displayed in the text fields. The user can then edit this information i.e. change the problem description or add more test cases. When the user is finished making their change they click save changes or else click cancel to discard their changes.

Coding Test Platform LOGOUT

String

Return Format

dcu ucd

String

Constraints

Input string less than 10000 characters

Sample Input

hello world!

Sample Output

ldrow olleh

Example with Explanation

hello world! reversed is ldrow olleh

SAVE CHANGES **CANCEL**

Figure 3.3

6.3 Add Challenges

If a user wants to add additional challenges to an already existing coding test, they simply click the add new challenge button (figure 3.4) from here they will be brought to the challenge setup form (figure 2.3) where they enter the information and test case for the additional challenges.



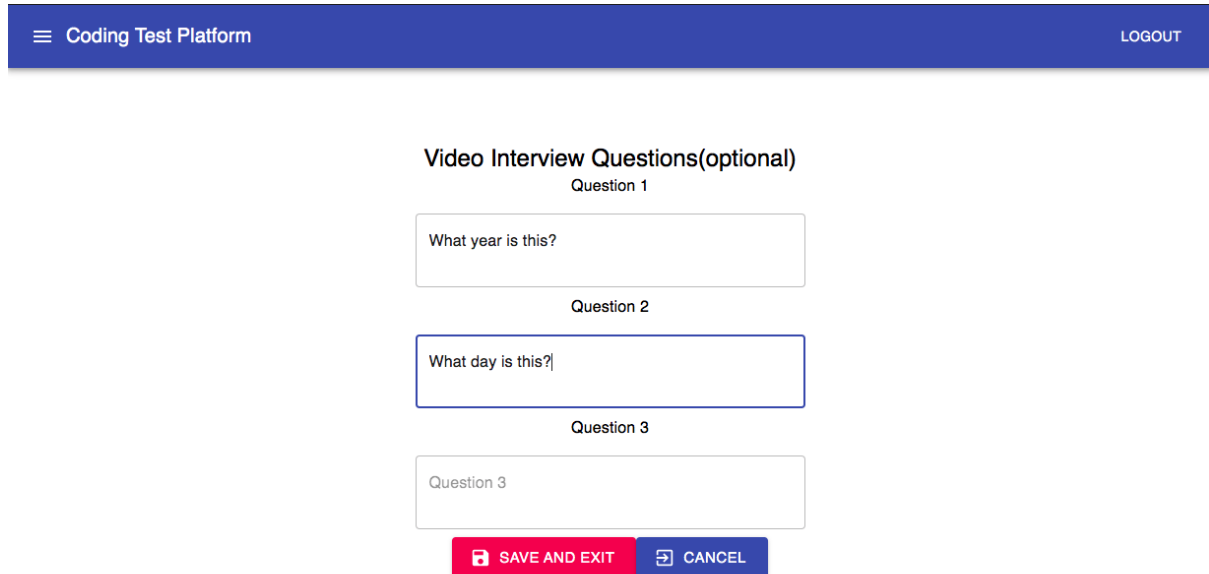
Title	Description	Date Created	ADD NEW CHALLENGE	
Reverse a string	Reverse a string from from sys.argv and print to stdout	2021-05-04		

Figure 3.4

6.4 Edit Questions

When a user clicks the edit icon for the questions in the edit test screen they will be brought to the edit questions form where all the current questions are displayed in the text fields. They can then edit, add or remove questions and click save and exit or else cancel to discard their changes.



Coding Test Platform LOGOUT

Video Interview Questions(optional)

Question 1

What year is this?

Question 2

What day is this?

Question 3

Question 3

SAVE AND EXIT CANCEL

Figure 3.5

6.5 Add Questions

If when creating the coding test the user didn't add any questions or else if they deleted the questions, they have the option to click the add questions button (figure 3.6) in the edit test screen. The user will then be brought to the add questions form (figure 2.5) where they can add questions to the coding test.



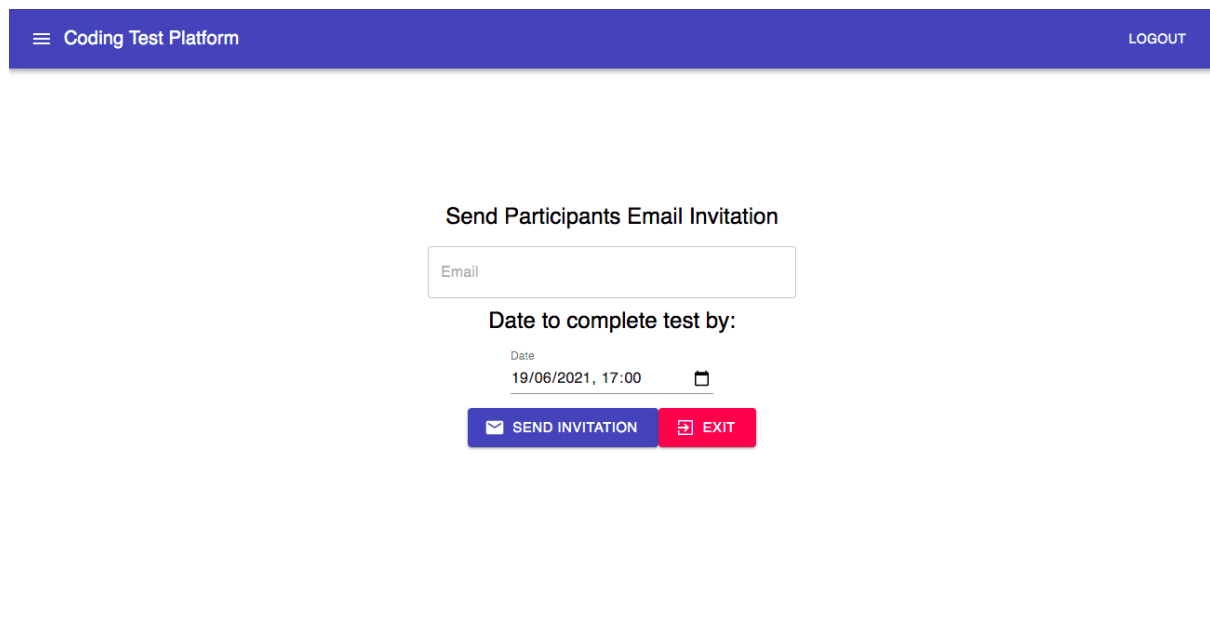
Question

ADD QUESTIONS

Figure 3.6

6.6 Add Participants

After clicking the add participants button in either the edit screen or the edit test screen, they will be directed to the send email invitations page (figure 3.7). Here the user enters the email of the participant they want to send an invitation to complete the coding test and also the date they expect them to complete the test by and then click send invitation. If the participant clicks the link in the email they receive after the expected date of completion they will not be allowed to attempt the test.



The screenshot shows a web interface for sending email invitations. At the top is a blue header bar with a hamburger menu icon, the text 'Coding Test Platform', and a 'LOGOUT' link. The main content area is titled 'Send Participants Email Invitation'. It contains a text input field labeled 'Email'. Below this is a label 'Date to complete test by:' followed by a date picker showing '19/06/2021, 17:00'. At the bottom are two buttons: a blue 'SEND INVITATION' button with an envelope icon and a red 'EXIT' button with a door icon.

Figure 3.7

If the email invitation is successfully sent an alert (figure 3.8) will be displayed to the user.

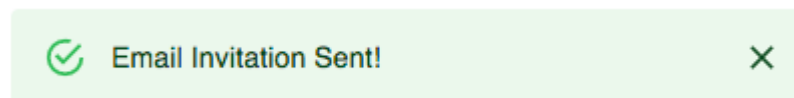


Figure 3.8

6.7 Reset Test

By clicking the reset test button (figure 3.9) in the edit screen (figure 3.1), this will remove all current results for that test from the database and any email invitations for that coding test not yet attempted by participants will be made invalid.



Test Name	Date Created			
My Coding test	2021-05-04	ADD PARTICIPANTS	RESET TEST	 

Figure 3.9

7. Results

After clicking on the view previous tests history/results card or the results button on the dashboard the user will be brought to the screen shown in figure 4.1 they will see a table displayed showing all their coding tests as well as the no of participants in that test and the number of challenges for that test. The user can click view to see more information about that coding test's results.

Previous Coding Tests Results				
Test Name	Date Created	No. of Participants	No. of Challenges	
My Coding test	2021-05-04T22:06:30.418Z	2	1	VIEW

Figure 4.1

7.1 Participants

After clicking the view button in the results page (figure 4.1), the user will see a list of the participants (email addresses) that were invited to attempt the coding test (figure 4.2). The user can click the view results button to view the individual results of each participant.

Coding Test Platform		LOGOUT
Coding Test Participants		
Participant		
mcnailypaul.33@gmail.com	VIEW RESULTS	
janeshk98@gmail.com	VIEW RESULTS	

Figure 4.2

7.2 Individual Results

After clicking the view results button in the participants list (figure 4.2) the user will be directed to the individual results page (figure 4.3). Here they will be able to view the percentage of test cases passed by the participant for each challenge and they can click the see more button to view more detailed results for each challenge. Also displayed are the interview questions for the coding test and the user clicks see response in order to watch the participants video response to each question asked.

Coding Test Platform		LOGOUT
Coding Test Results for: janeshk98@gmail.com		
Coding Challenges		
Title	Test Cases Passed	
Sum Digits	100%	SEE MORE
Reverse a string	100%	SEE MORE
Question Responses		
No.	Question	
1	What is your name?	SEE RESPONSE
2	What year is this?	SEE RESPONSE
3	What time is it?	SEE RESPONSE

Figure 4.3

7.3 Challenge Results

If the user clicks the see more button for a challenge in the individual results page (figure 4.3). They will be brought to a more detailed results page for that particular challenge (figure 4.4). On this page the user can see the participants code solution for the challenge as well as the output for each test case, the runtime and memory for each test case.

Coding Test Platform						LOGOUT
Coding Challenge Results for: Reverse a string						
Code for Solution:						
<pre>1 import sys 2 3 for line in sys.argv[1:]: 4 print(line[::-1])</pre>						
Input	Expected Output	Actual Output	Passed?	Runtime	Memory	
hello	olleh	olleh	true	0.442 seconds	30736 K/B	
123	321	321	true	0.418 seconds	30768 K/B	
DCU	UCD	UCD	true	0.405 seconds	30928 K/B	
code	edoc	edoc	true	0.508 seconds	31024 K/B	
69	96	96	true	0.372 seconds	30752 K/B	

Figure 4.4

7.4 Video Responses

To view a participants response to a question the user clicks the see response button in the individual results page (figure 4.3). They will then see a pop up card with a video player that will play the participants recorded and saved response to that particular question.

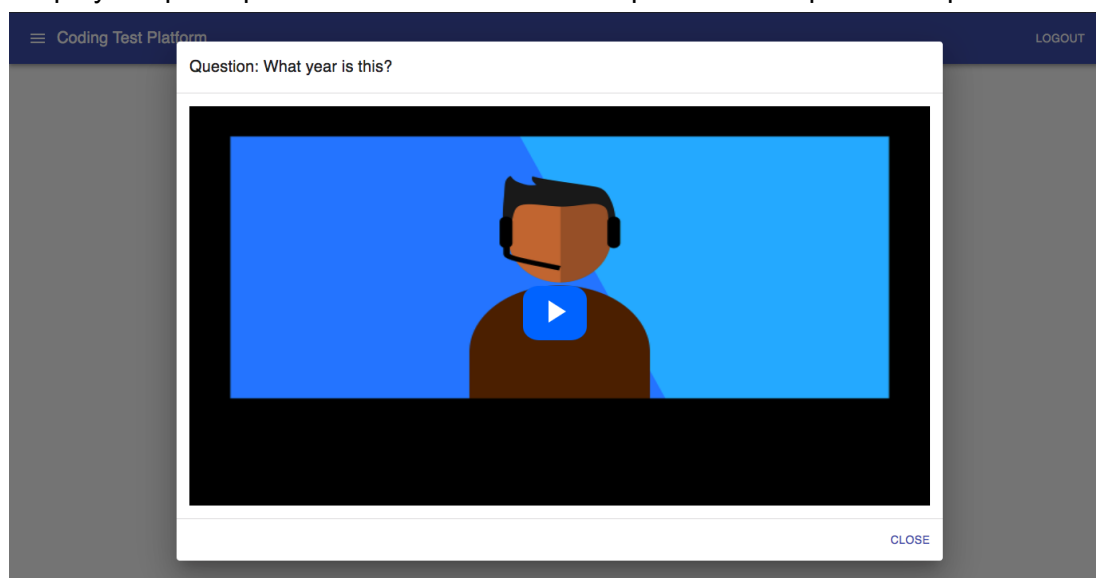


Figure 4.5

8. Coding Test

A participant who is invited to attempt a coding test will receive an email invitation with a unique link, when the participant clicks the link they will be brought to the coding test (figure 5.1). Here they will see the coding challenge description on the left side of the screen which explains what they are expected to write a solution for. The participant can select which language they will be writing their solution in in the header bar, the header bar also shows the time remaining to complete the test. They must write their solution in the text editor on the right side of the screen. The bottom right of the screen shows the test cases for the challenge and when the participant clicks the run code button to run their code they will be able to see which of the test cases passed and/or which of the test cases failed. When the participant has finished writing their solution to all of the coding challenges they have to click the submit test button in the top right corner of the screen and their coding test solution and results will be saved in the database.

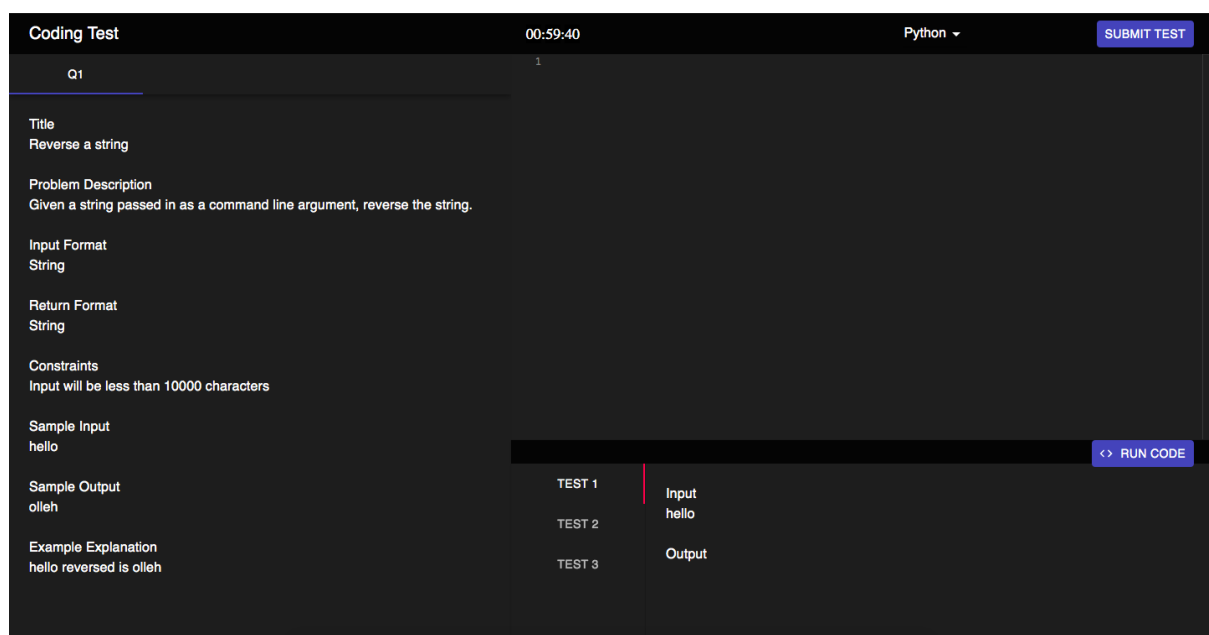


Figure 5.1

9. Video Interview

After a participant clicks the submit test in the coding test (figure 5.1), if the coding test has interview questions they will be brought to the video interview screen (figure 6.1) (A webcam will be required). The question asked will be displayed on the left side of the screen. To begin recording the video they click the start recording button.

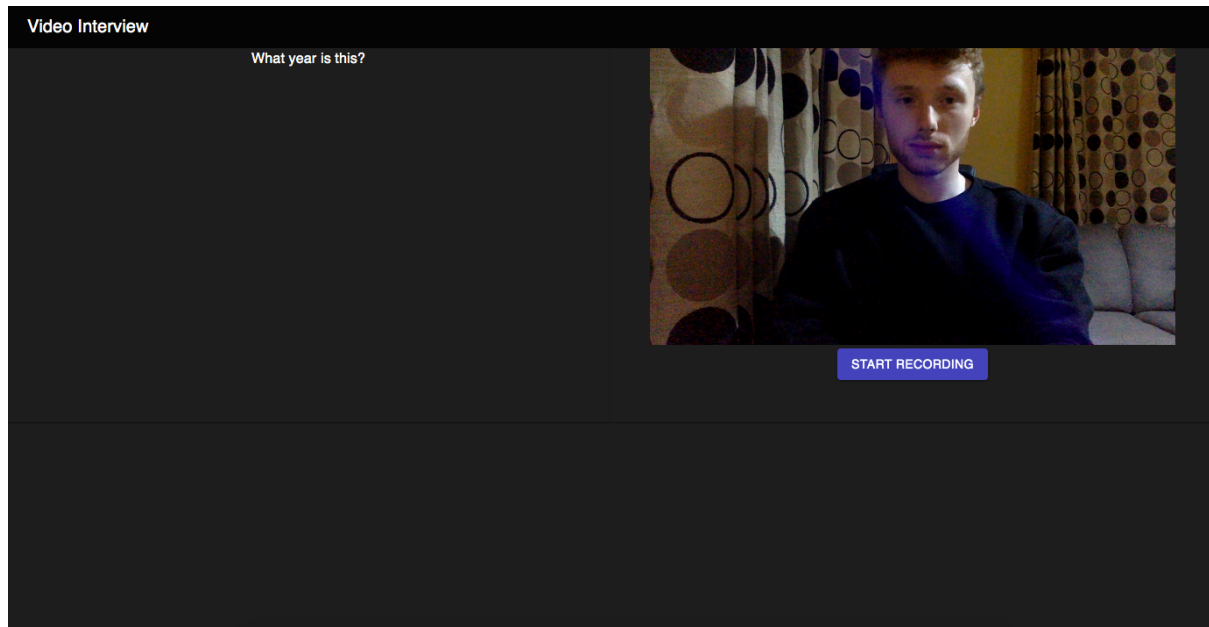


Figure 6.1

When recording the participant will have 60 seconds to answer the question, they can either take the full 60 seconds to answer the question or else click the stopwatch recording button when they are finished answering the question.

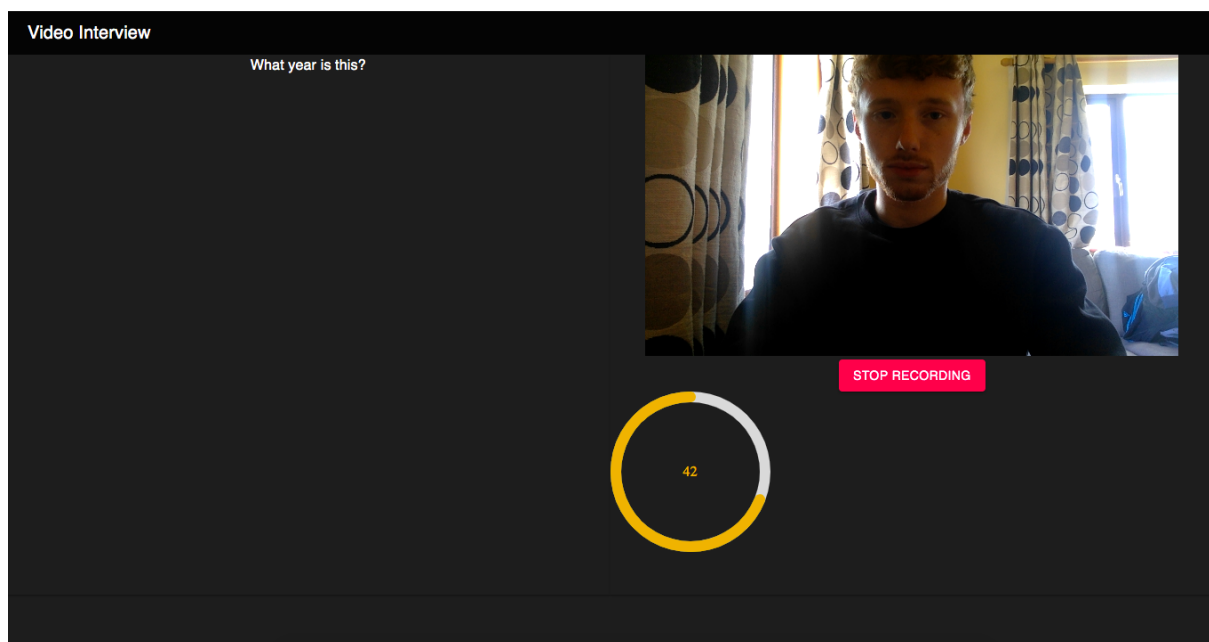
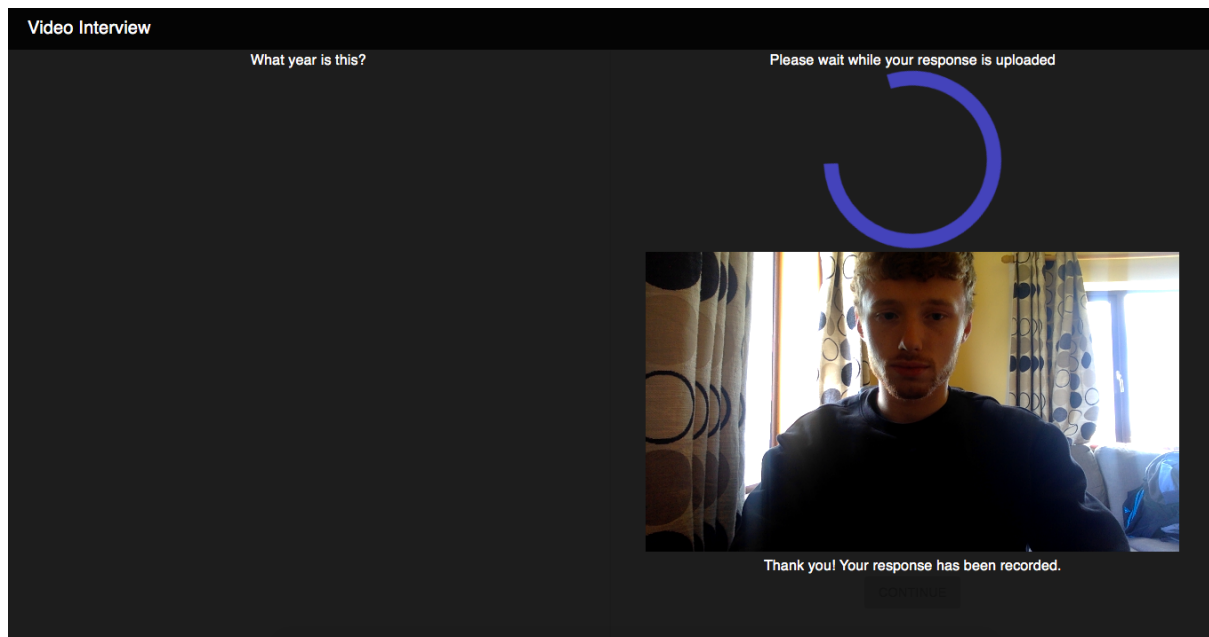


Figure 6.2

After recording has stopped the participant will be asked to wait while their response is being uploaded (figure 6.3), this is to ensure that they don't close the window before it is finished being uploaded.



Once the video interview is completed and all responses have been uploaded to storage the participant will see the screen in figure 6.4 and they can then close the window to exit the video interview.

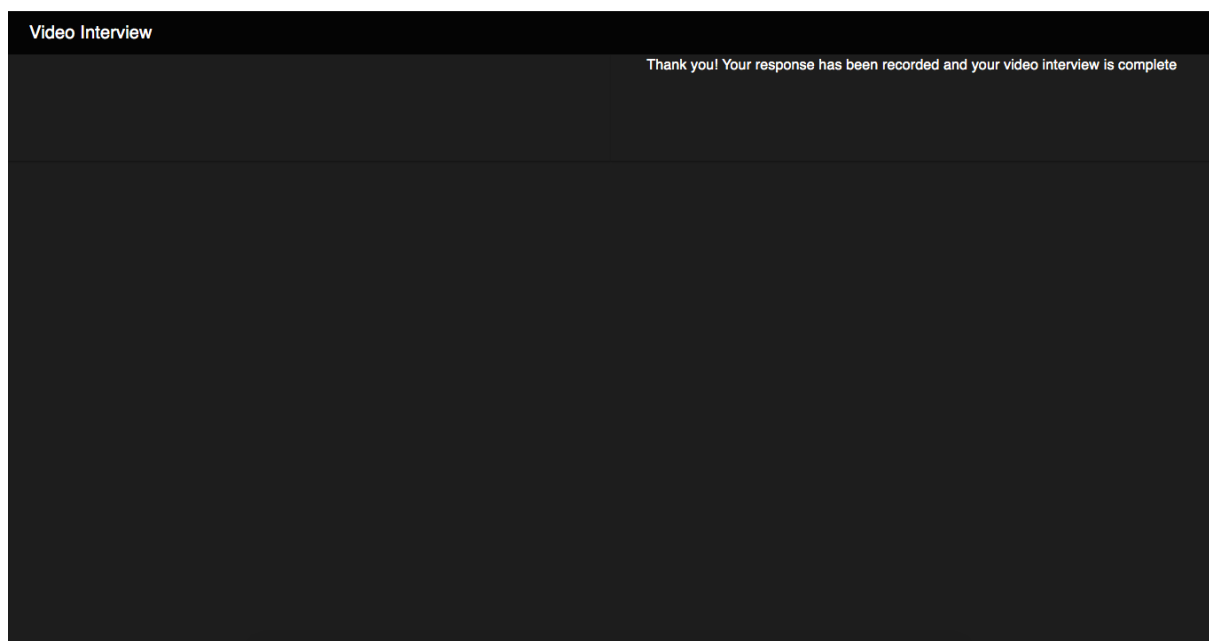


Figure 6.4