Talent in Innovation.
Innovation in Talent.

SHL Release notes

November 2021

Version: 1.0 | Last updated: 6 December 2021

Talent in Innovation.
Innovation in Talent.

TalentCentral

Release Notes

Summary – 30th November Release

On 30th November – 1st December, the following feature improvements and bug fixes are planned to be released into the TalentCentral platform.

In this Release:

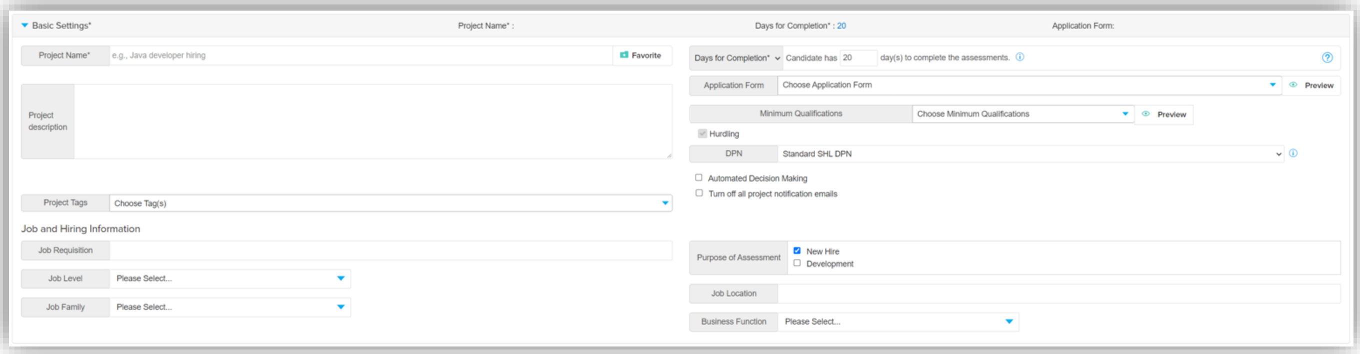
TalentCentral

'Job Info' data in Create and Manage Project

We have introduced a new 'Job Info' section to the TalentCentral project configuration process.

This is an optional section where users can provide context information for why they are using the platform to assess participants. The information that can be captured is:

- **Job Requisition:** a free text field for users to enter the job requisition identifier.
- **Job Level:** a list of generic seniority levels, users are expected to select the closest fit.
- **Job Family:** a list of job discipline types, users are expected to select the closest fit.
- **Purpose of Assessment:** a choice between New Hire or Development.
- **Job Location:** a free text field for users to enter the location of the job (limited to 100 characters).
- **Business Function:** a list of generic business functions, users are expected to select the closest fit.



Basic Settings* | Project Name* : | Days for Completion* : 20 | Application Form:

Project Name*: e.g., Java developer hiring | Favorite

Project description:

Project Tags: Choose Tag(s)

Job and Hiring Information

Job Requisition:

Job Level: Please Select...

Job Family: Please Select...

Days for Completion*: Candidate has 20 day(s) to complete the assessments.

Application Form: Choose Application Form | Preview

Minimum Qualifications: Choose Minimum Qualifications | Preview

☑ Hurdling

DPN: Standard SHL DPN

□ Automated Decision Making

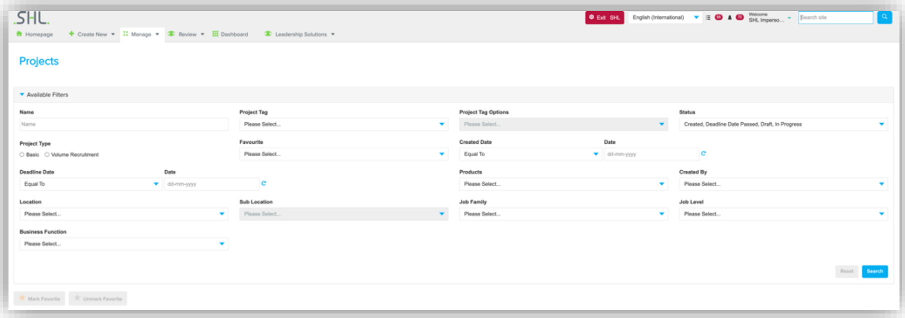
□ Turn off all project notification emails

Purpose of Assessment: ☑ New Hire □ Development

Job Location:

Business Function: Please Select...

This information helps users elsewhere in the application to find projects through the search filters in the Manage and Review areas of the platform.



Projects

Available Filters

Name: Name

Project Tag: Please Select...

Project Tag Options: Please Select...

Status: Created, Deadline Date Passed, Draft, In Progress

Project Type: ○ Basic ○ Volume Recruitment

Favourite: Please Select...

Created Date: Equal To | Date: dd-mm-yyyy

Deadline Date: Equal To | Date: dd-mm-yyyy

Products: Please Select...

Created By: Please Select...

Location: Please Select...

Sub Location: Please Select...

Job Family: Please Select...

Job Level: Please Select...

Business Function: Please Select...

Reset | Search

Mark Favorite | Unmark Favorite

Client Analytics

Product Overview

Client Analytics is a new analytics module available via the TalentCentral platform that will enable our clients to track Key Performance Indicators (KPIs) with respect to their Talent Acquisition programs. It provides actionable insights about your assessment programs, allowing you to easily see trends and identify opportunities and/or problem areas. These insights help you to optimize your recruitment efforts, measure the impact of previous recruitment programs and plan for future programs in a more efficient way.
This new analytics module will be available in place of your existing TalentCentral Dashboard.

Client Analytics dashboards can be utilized to address various use cases such as:

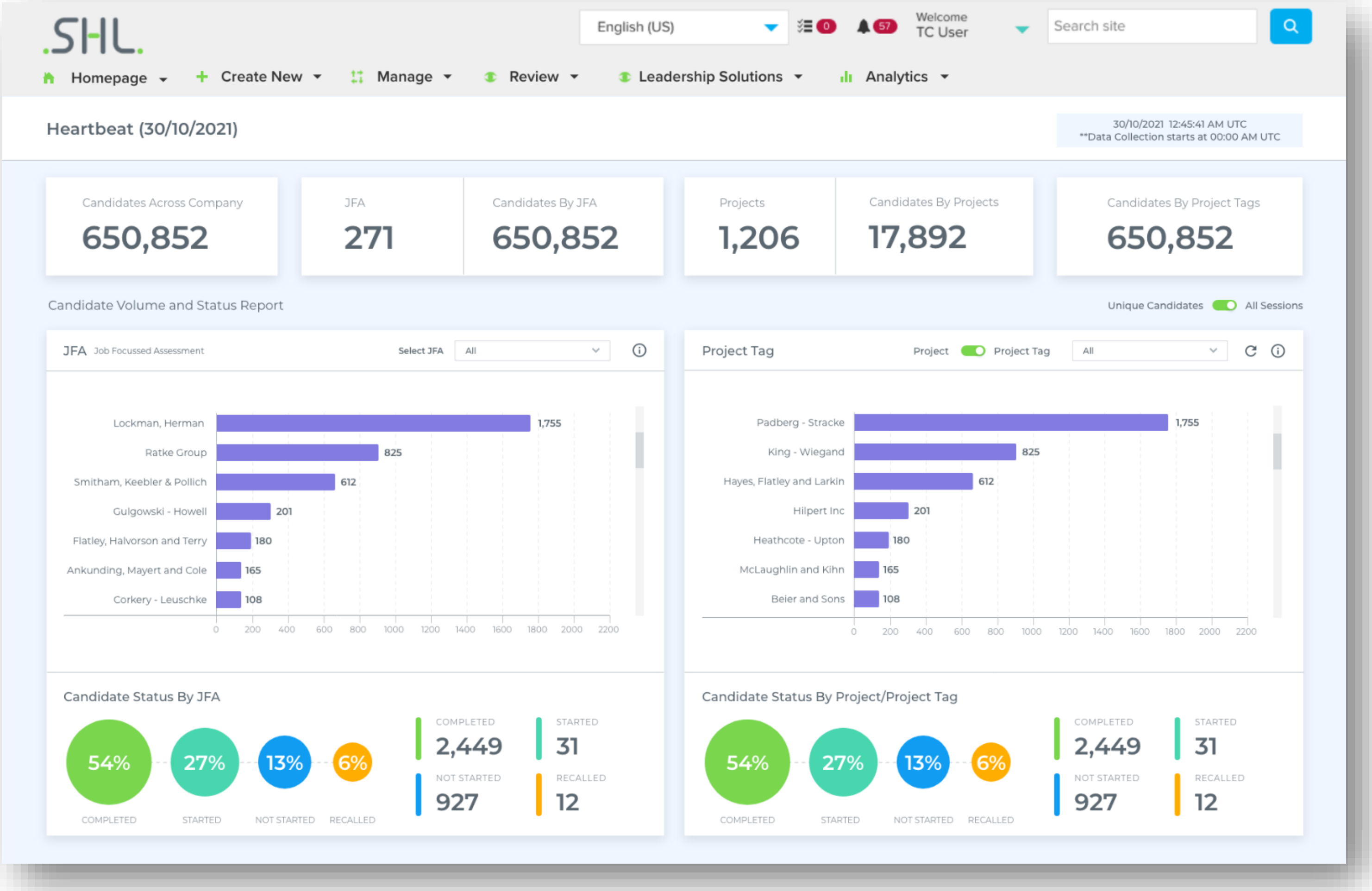
- Understand Year on Year (YoY) performance of assessment programs.
- Are we getting the right number of candidates?
- Are enough candidates passing?
- How is drop-out trending, are adjustments to the candidate experience needed?
- How are candidates performing on different competencies – Verify, MQ, Leadership, UCF, Sales?
- Get real time insights of live recruitment campaign(s) e.g., track completion rate – low completion rates might be an indicator that there is a need to share an extra nudge/reminder to candidates to take the assessments.
- How is assessment experience on different devices? E.g., are we providing uniform experience to candidates across all devices or are there obvious differences that should be addressed?

Client Analytics Dashboards

There are 5 analytics dashboards available including: Heartbeat, Pipelines, JFA Summary, Project Summary, and Volume Tracking

Heartbeat

Near real-time monitoring of active projects or events to ensure opportunities and/or problems are identified sooner.



Heartbeat (30/10/2021)

30/10/2021 12:45:41 AM UTC
**Data Collection starts at 00:00 AM UTC

Candidates Across Company 650,852

JFA 271

Candidates By JFA 650,852

Projects 1,206

Candidates By Projects 17,892

Candidates By Project Tags 650,852

Candidate Volume and Status Report

JFA Job Focussed Assessment

Project Tag

Candidate Status By JFA

COMPLETED 2,449

STARTED 31

NOT STARTED 927

RECALLED 12

Candidate Status By Project/Project Tag

COMPLETED 2,449

STARTED 31

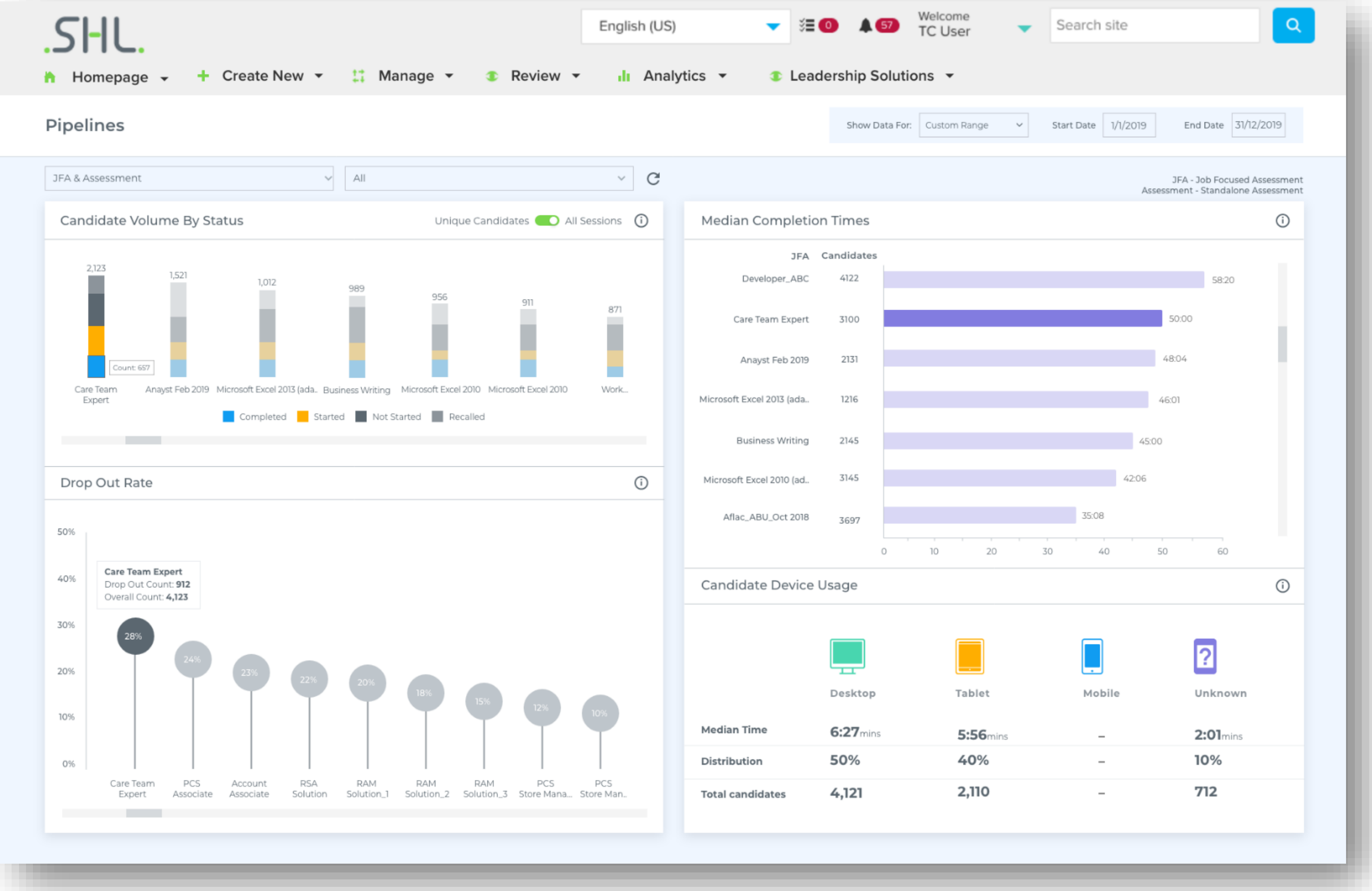
NOT STARTED 927

RECALLED 12

Figure 1. Heartbeat Dashboard

Pipelines

- Analyse the effectiveness and efficiency of recruitment programs to meet business objectives regarding ‘identifying the right talent in the right time’
- Standardized data visualizations focused on Job-Focused Assessments (JFAs) and/or Standalone Assessments (Verify, OPQ32 etc.)

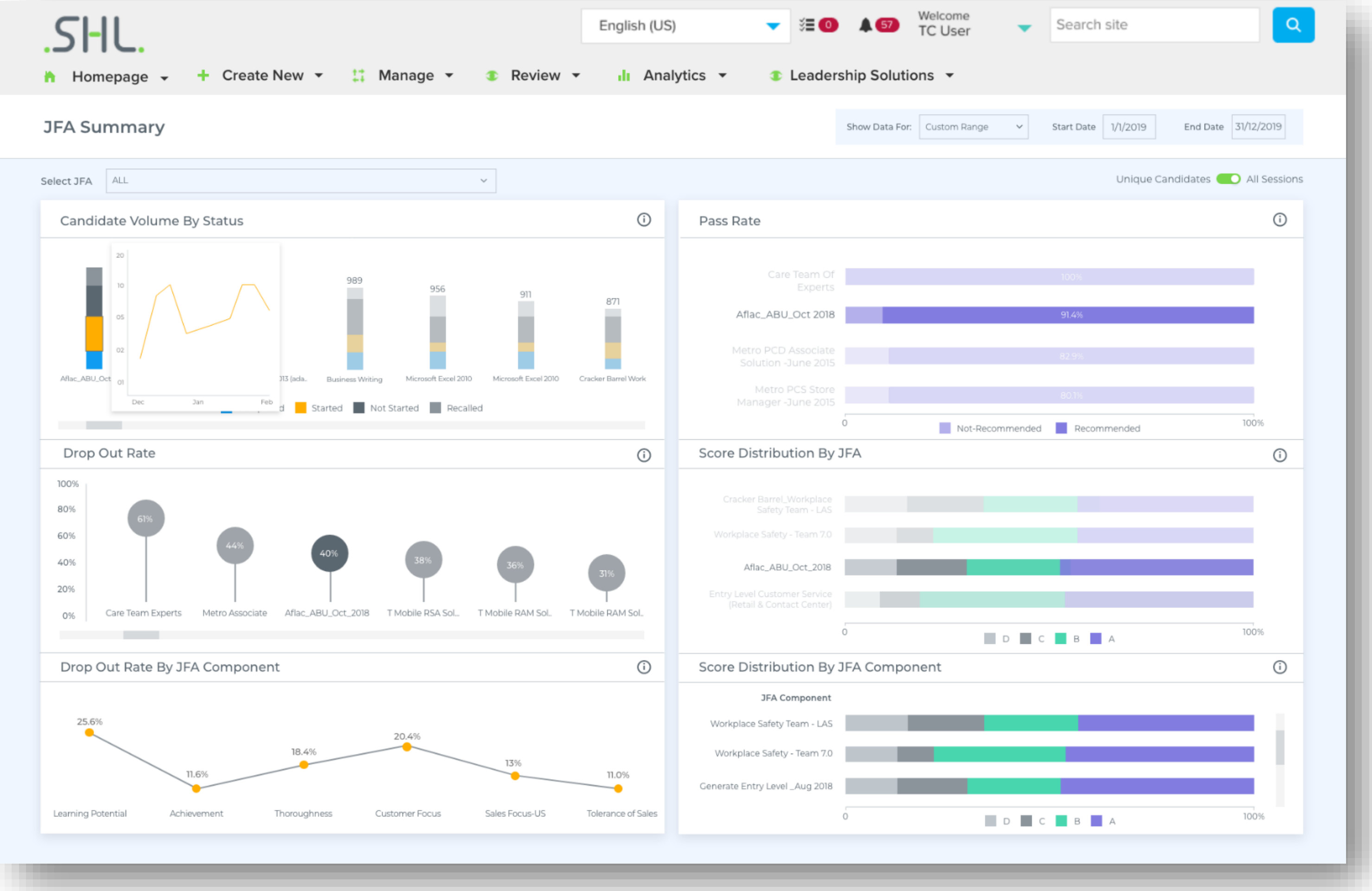


	Desktop	Tablet	Mobile	Unknown
Median Time	6:27mins	5:56mins	–	2:01mins
Distribution	50%	40%	–	10%
Total candidates	4,121	2,110	–	712

Figure 2. Pipelines Dashboard

JFA Summary

- Analyse the effectiveness and efficiency of Job-Focused Assessments (JFAs) to meet business objectives regarding ‘identifying the best talent’.
- Standardized data visualizations focused only on JFAs.



English (US)

Welcome TC User

Search site

JFA Summary

Show Data For: Custom Range

Start Date 1/1/2019

End Date 31/12/2019

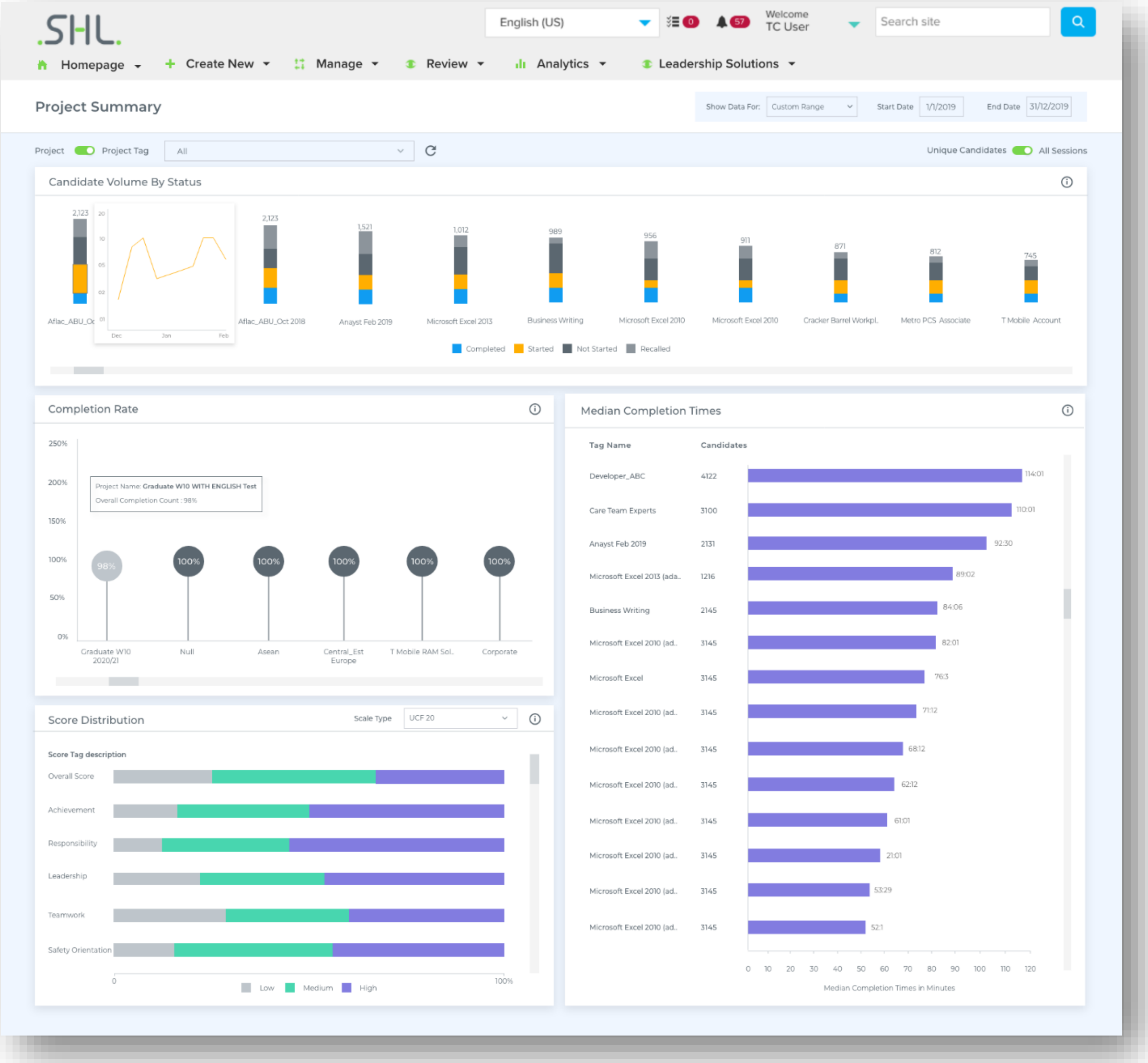
Select JFA ALL

Unique Candidates All Sessions

Figure 3. JFA Summary Dashboard

Project Summary

- Analyse the effectiveness and efficiency of Projects to meet business objectives regarding ‘identifying the best talent’.
- Standardized data visualizations focused on aggregating and visualizing data across Projects or Project Tags.



Show Data For: Custom Range

Start Date: 1/1/2019

End Date: 31/12/2019

Project Tag: All

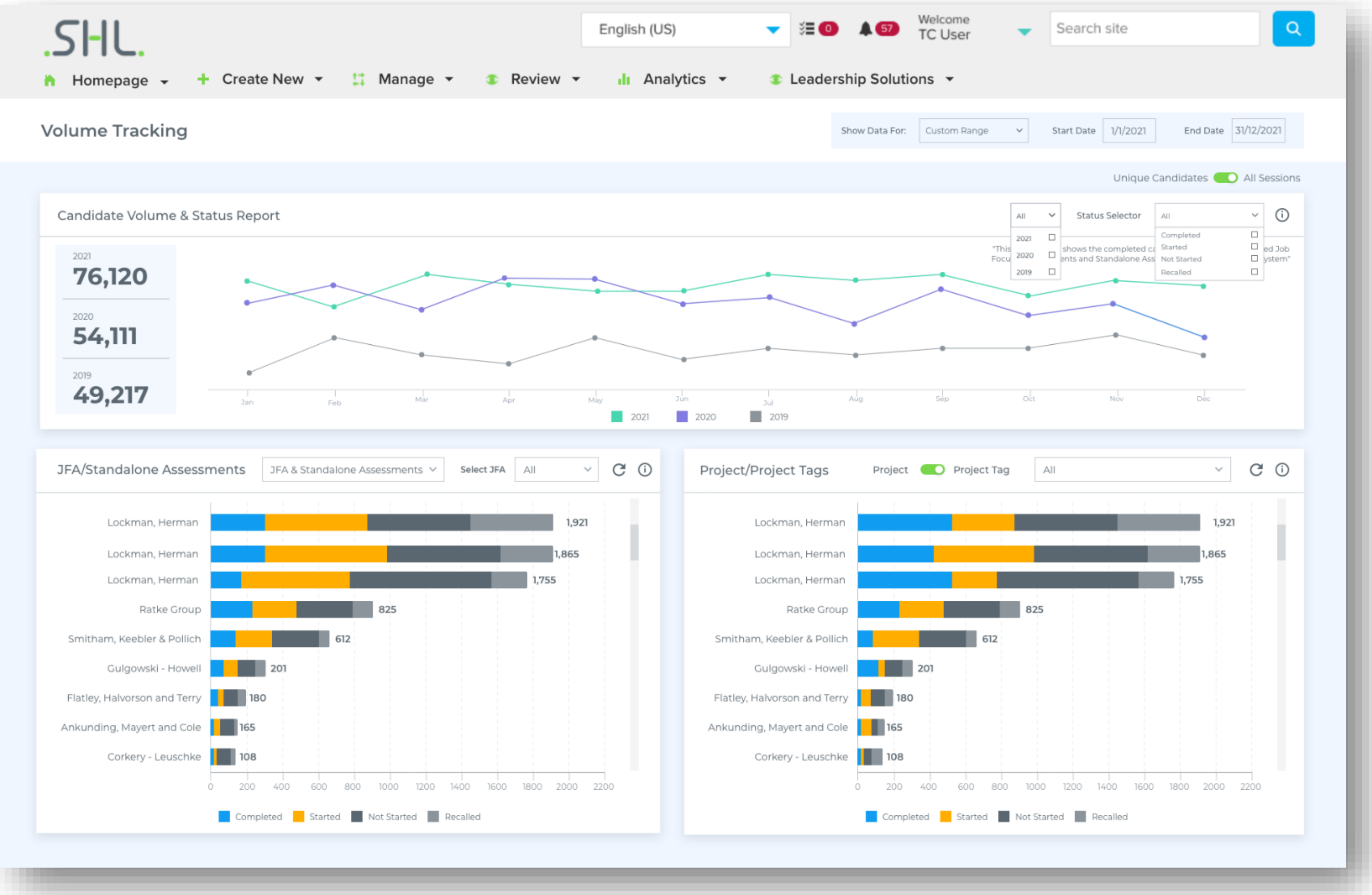
Scale Type: UCF 20

Tag Name	Candidates	
Developer_ABC	4122	114:01
Care Team Experts	3100	110:01
Anayst Feb 2019	2131	92:30
Microsoft Excel 2013 (ada..	1216	89:02
Business Writing	2145	84:06
Microsoft Excel 2010 (ad..	3145	82:01
Microsoft Excel	3145	76:3
Microsoft Excel 2010 (ad..	3145	71:12
Microsoft Excel 2010 (ad..	3145	68:12
Microsoft Excel 2010 (ad..	3145	62:12
Microsoft Excel 2010 (ad..	3145	61:01
Microsoft Excel 2010 (ad..	3145	21:01
Microsoft Excel 2010 (ad..	3145	53:29
Microsoft Excel 2010 (ad..	3145	52:1

Figure 4. Project Summary Dashboard

Volume Tracking

- Analyse the participant groups coming through specific recruitment programs over time.
- Standardized data visualizations focused on Job-Focused Assessments (JFAs), Standalone Assessments, Projects, and Project Tags



Show Data For: Custom Range

Start Date 1/1/2021

End Date 31/12/2021

Figure 5. Volume Tracking Dashboard

Key Performance Indicators (KPIs)

Client Analytics dashboards can be used to track the following KPIs:

- **Candidate Volume -** Number of candidates across an assessment or project grouped by status: Completed, Started, Not Started, Recalled.
- **Drop Out Rate -** Percentage of candidates that started an assessment or project but did not complete it.
- **Pass Rate -** Percentage of candidates that completed an assessment and passed i.e., reached a pre-configured cut off score.
- **Score Distribution -** Percentage distribution of candidates falling in each of the pre-configured score zones for an assessment or project.
- **Completion Time-** Median time spent by candidates to complete an assessment or project.
- **Device Usage-** Percentage distribution of candidates across different device types i.e., Desktop, Mobile, Tablet

Recommended Screen Size & Resolution

Screen Size (Desktop/Laptop)	Screen Resolution
13”	1366 x 768
15”	1366 x 768

Access

Client Analytics dashboards will be available only in TalentCentral US, AU and EU regions at present. Please contact your account manager to find out more details and get access to this new analytics module.

Smart Interview Live

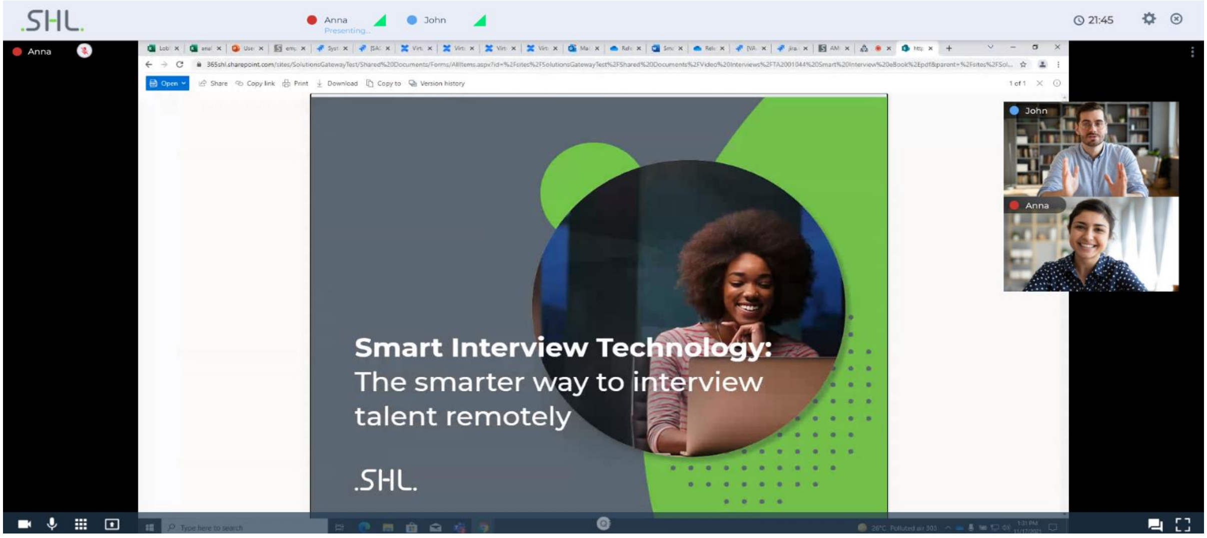
New Features

Screen share - We are introducing the capability to screen share during interviews in Smart Interview Live and VADC. Both candidates and participants will be able to share their entire screen, tab or any window and other participants would be able to see the shared screen.
Note: Users can only share their entire screen if using safari browser.

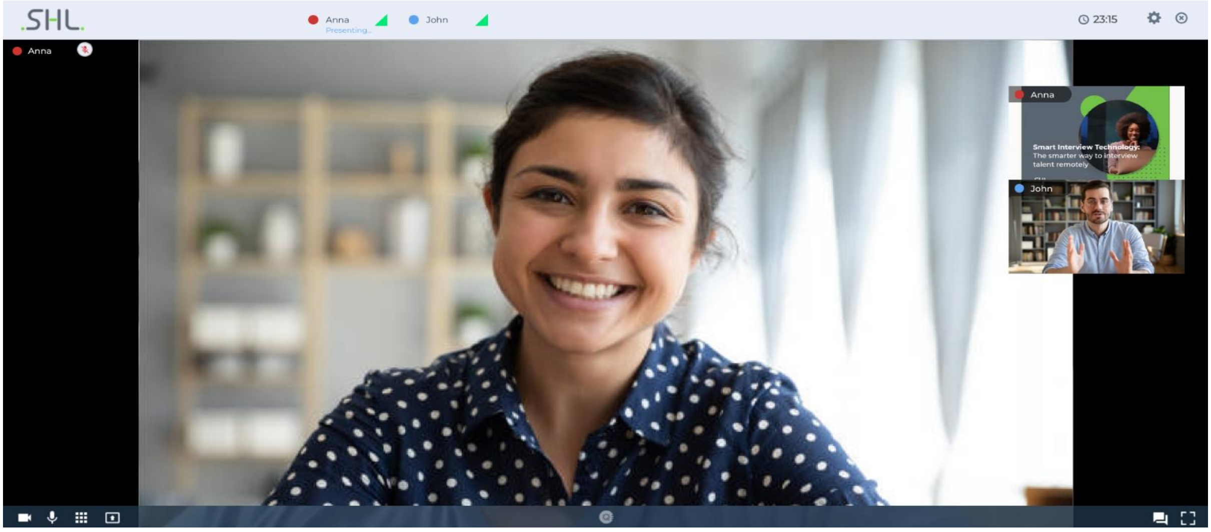
Participants can also use other built in features to change the screen view:

a. **Pin**: Participants viewing the shared screen can pin any participant video to see that participant instead of the shared screen in the bigger frame. The shared screen will be visible with the rest of the participants' videos.

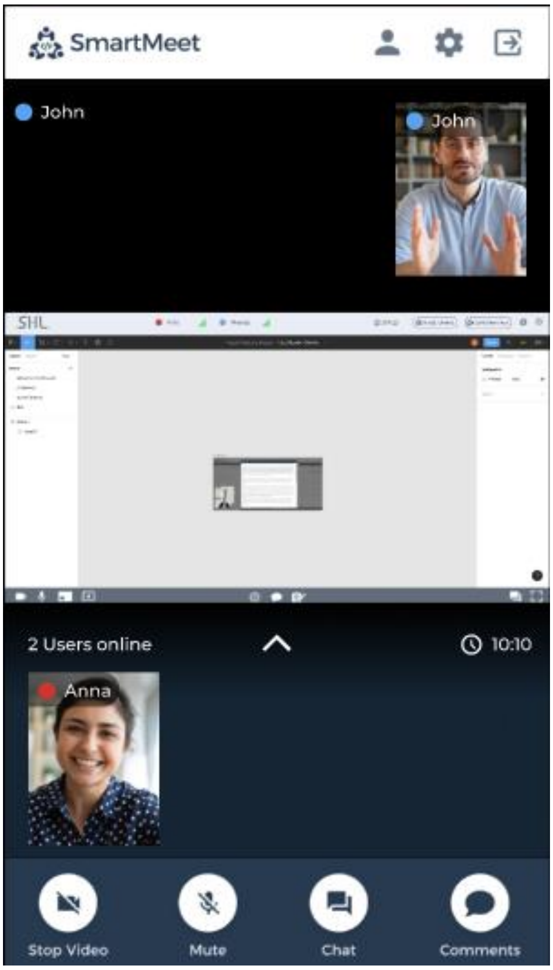
b. **Grid View**: The shared screen is also visible in grid view along with other participants' videos.

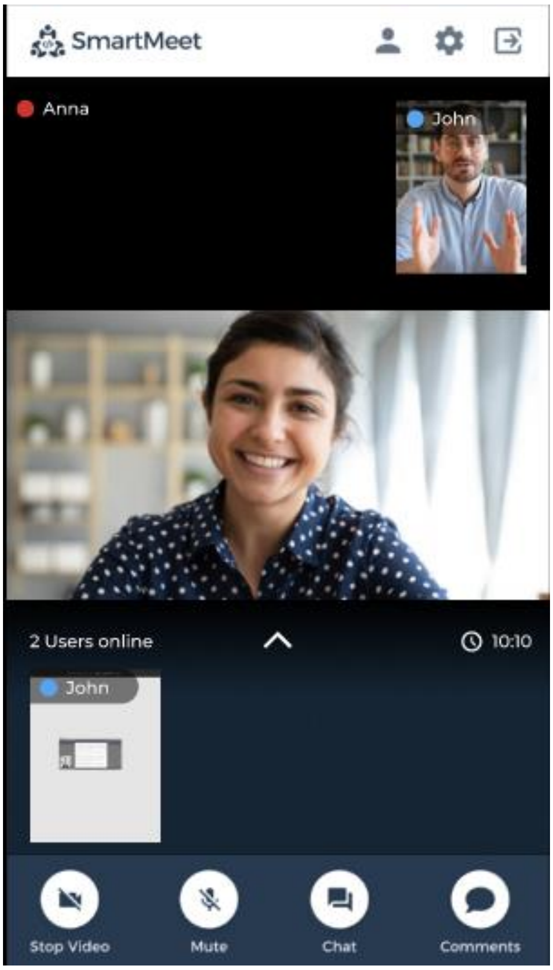


Smart Interview Technology:
The smarter way to interview talent remotely



Screen share for mobile app (for iAssess users only) – Participants can view the shared screen on the app but will not be able to share their screen from the app at the moment.





Note: The shared screen will not be visible in the playback/recording with this release. We will do a follow up release in early Q1 after which the screen shared during the interview would also be visible in playback.

VADC

TC VADC user can download the candidate document

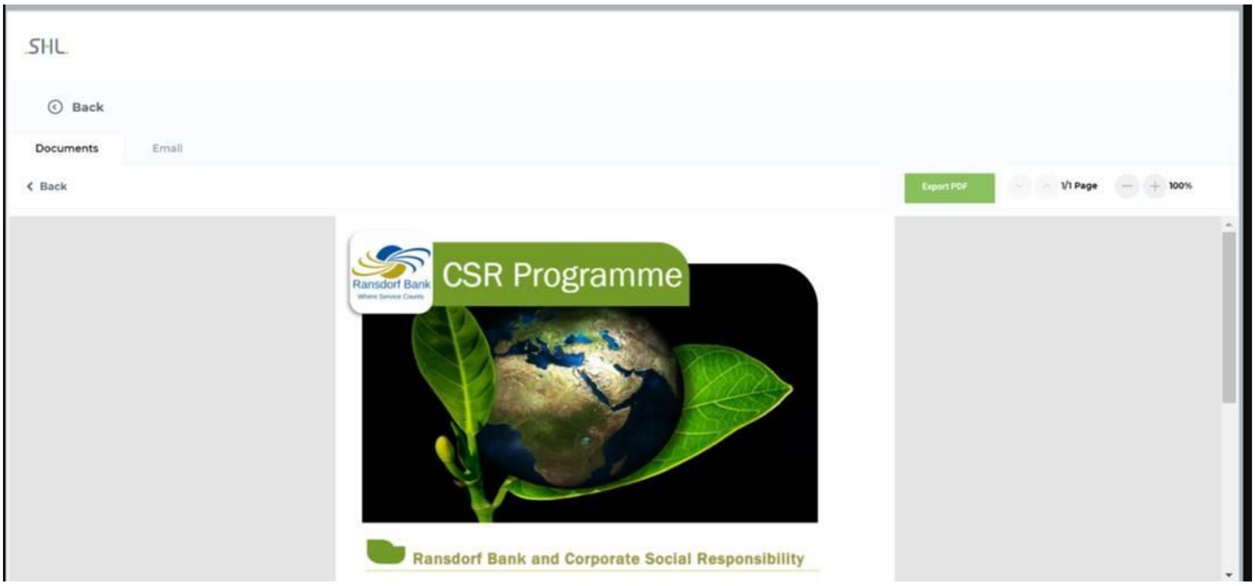
Overview

The assessor/TC user can download the candidate document that was uploaded in the VADC exercise.

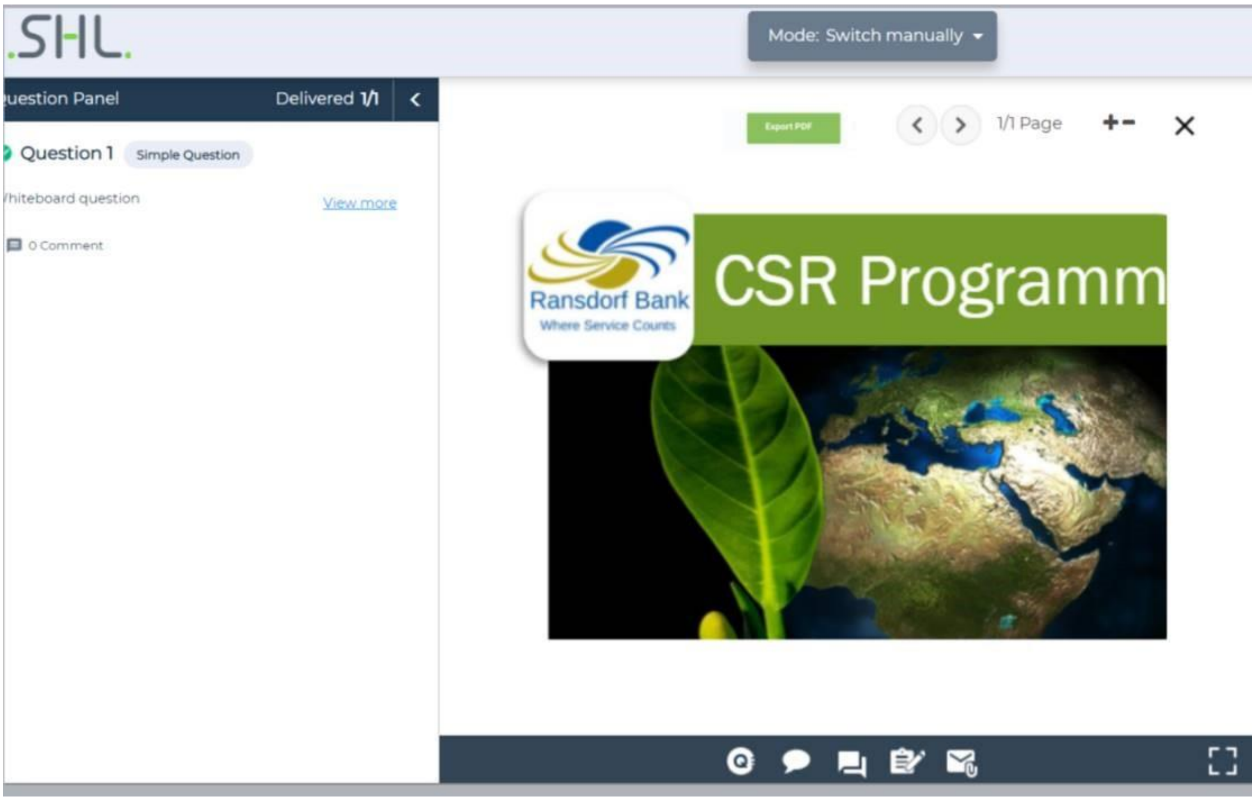
- The assessor can download the candidate documents from the TC review assessor dashboard.
- Only the document(s) uploaded by the candidate can be downloaded. This means, candidate email responses and/or original exercise emails and/or the documents uploaded as part of email content cannot be downloaded.
- The uploaded document can only be downloaded in PDF format. This means even if the candidate uploads a Word and/or PPT and/or Excel, it will be downloaded in PDF format.

How can the assessor/TC user view the uploaded candidate document?

- Post exercise completion, the TC assessor can navigate to the TC review dashboard and click on the response document uploaded by the candidate.
- The response document uploaded by the candidate is opened in a PDF preview with an "Export PDF" button.



- Clicking on the "Export PDF" button will download the candidate document in PDF format.
- The export PDF option is now available for the VADC inbox simulation and the live meeting replay link.



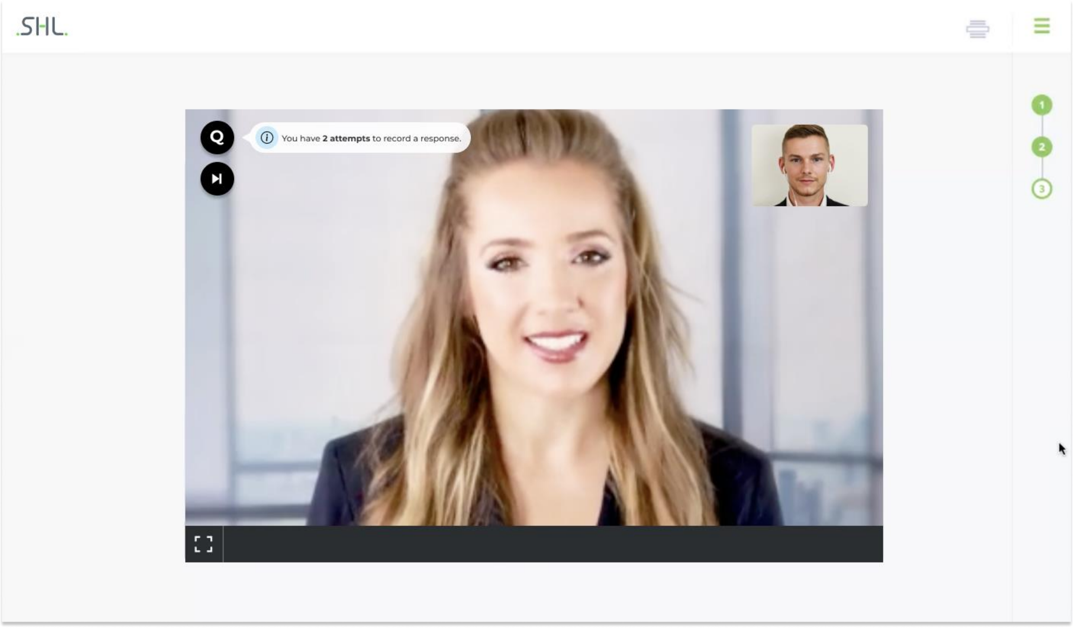
Note - Any user who has access to the TC project containing the VADC exercise can navigate to the assessor dashboard, open the document preview, and export the PDF.

Smart Interview on Demand

Re-record response & question container fixes: Releasing across all regions by 1st week of December 2021. These will be rolled out for all clients automatically.

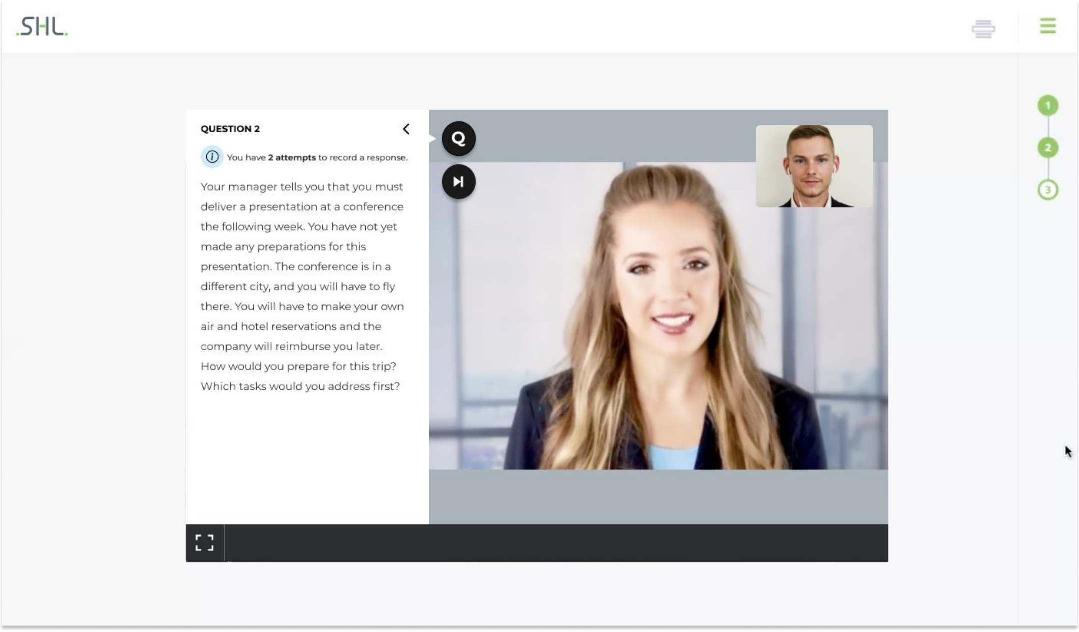
- Re-record: Functionality for a candidate to re-record their responses while attempting the questions in a video interview. The number of attempts on a question can be configured by the recruiter on a question level and can then be modified while defining a question set for the interview. These attempts are what will be available for the candidates to utilize during their response time.
- Question container fixes: Currently the question text container often overlaps with the interviewer video during question delivery. We have now introduced a split screen approach to ensure the media being played is unaffected by the question text container.

Screenshots from the participant experience below:



You have **2 attempts** to record a response.

Figure 6 Question container closed

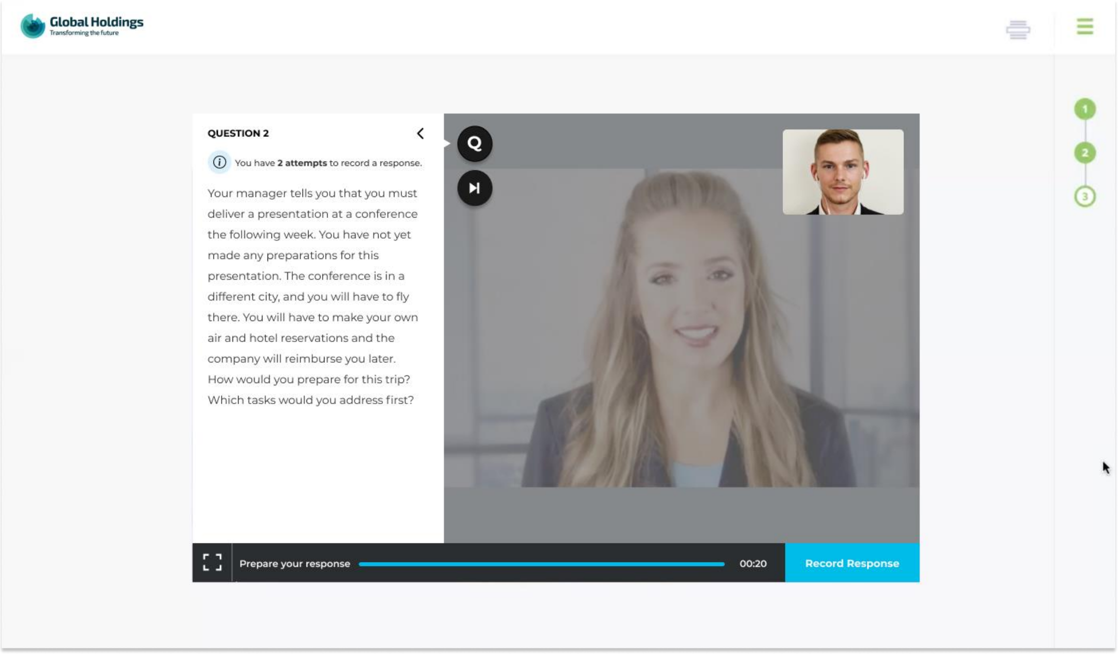


QUESTION 2

You have **2 attempts** to record a response.

Your manager tells you that you must deliver a presentation at a conference the following week. You have not yet made any preparations for this presentation. The conference is in a different city, and you will have to fly there. You will have to make your own air and hotel reservations and the company will reimburse you later. How would you prepare for this trip? Which tasks would you address first?

Figure 7 Question container open



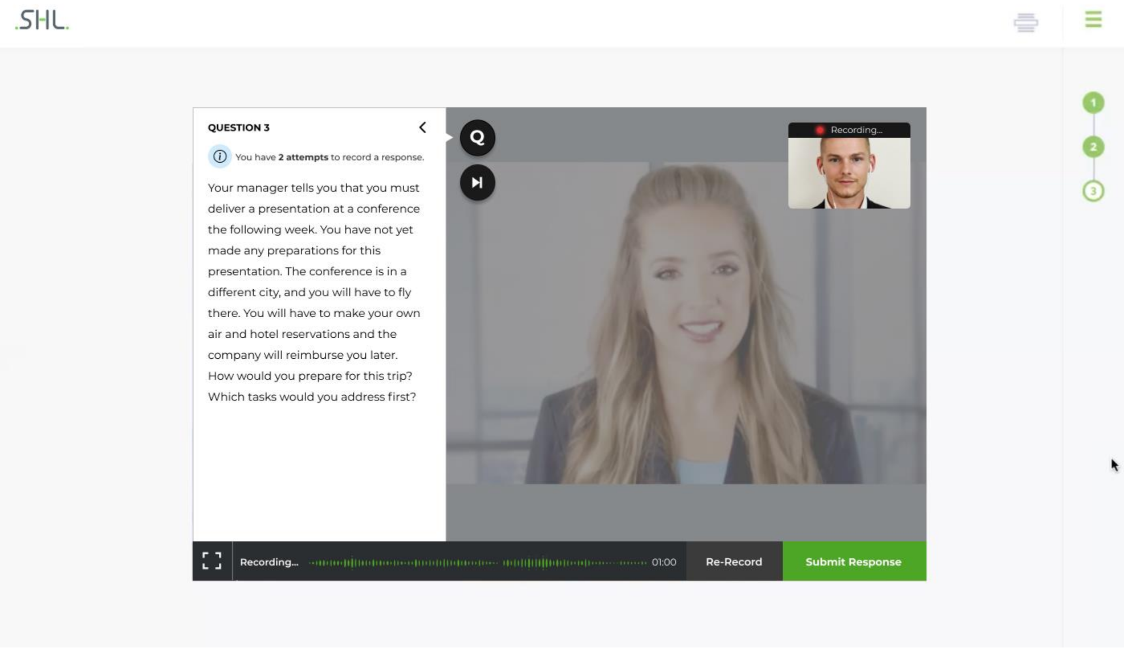
QUESTION 2

You have **2 attempts** to record a response.

Your manager tells you that you must deliver a presentation at a conference the following week. You have not yet made any preparations for this presentation. The conference is in a different city, and you will have to fly there. You will have to make your own air and hotel reservations and the company will reimburse you later. How would you prepare for this trip? Which tasks would you address first?

Prepare your response 00:20

Figure 8 Preparation time with attempt info



QUESTION 3

You have **2 attempts** to record a response.

Your manager tells you that you must deliver a presentation at a conference the following week. You have not yet made any preparations for this presentation. The conference is in a different city, and you will have to fly there. You will have to make your own air and hotel reservations and the company will reimburse you later. How would you prepare for this trip? Which tasks would you address first?

Recording... 01:00

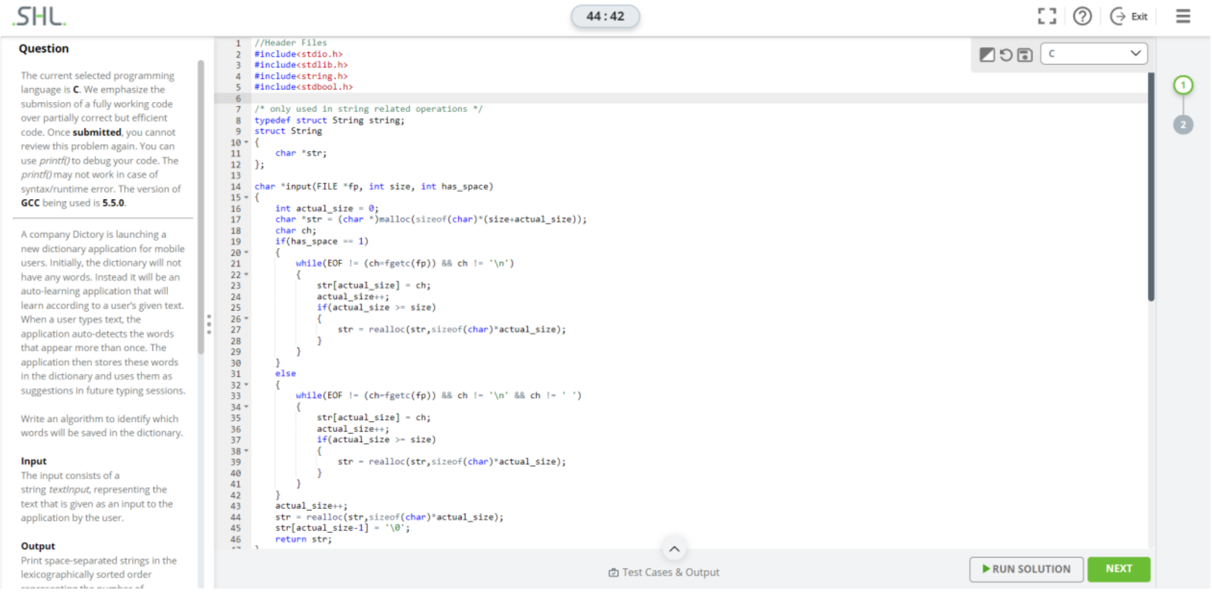
Figure 9 Response time with re-record button with attempts available

Coding & Simulations

Improvement: Automata UI/UX refresh

The Automata interface has been refreshed to meet the latest standards and make it more realistic while attempting the coding assessment. A summary of changes is outlined below, along with the current interface:

- The editor size will be increased to give the candidate more space to write the code.
- The coding assessment-specific controls/actions such as save, dark mode, and reset will be shown inside the editor along with the language switch options.
- The "Test cases & Output" section will move to the footer and the "Editor" tab will be removed.
    - The test cases window will have expand/collapse controls.
    - The candidate will be able to view the sample test cases as well as access to create custom test cases in this section.
- The "Compile and Run" button will be changed to "Run Solution" and move to the footer (below the editor).
- We are also changing the full-screen functionality where clicking on the full-screen icon will take the editor to the full screen along with the header, footer, and question in the minimized state so that the candidate will have all the essential controls handy in the full-screen mode as well (unlike the previous interface).
- For coding assessments, "Next" will be shown instead of "Submit Response" wherever applicable and remains unchanged if not applicable.
- Candidates can also resize the question panel at any time by dragging the 3 dot icon. A single click on it will minimize the question panel.



Question

The current selected programming language is **C**. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use *printf()* to debug your code. The *printf()* may not work in case of syntax/runtime error. The version of **GCC** being used is **5.5.0**.

A company Dictory is launching a new dictionary application for mobile users. Initially, the dictionary will not have any words. Instead it will be an auto-learning application that will learn according to a user's given text. When a user types text, the application auto-detects the words that appear more than once. The application then stores these words in the dictionary and uses them as suggestions in future typing sessions.

Write an algorithm to identify which words will be saved in the dictionary.

Input

The input consists of a string *textInput*, representing the text that is given as an input to the application by the user.

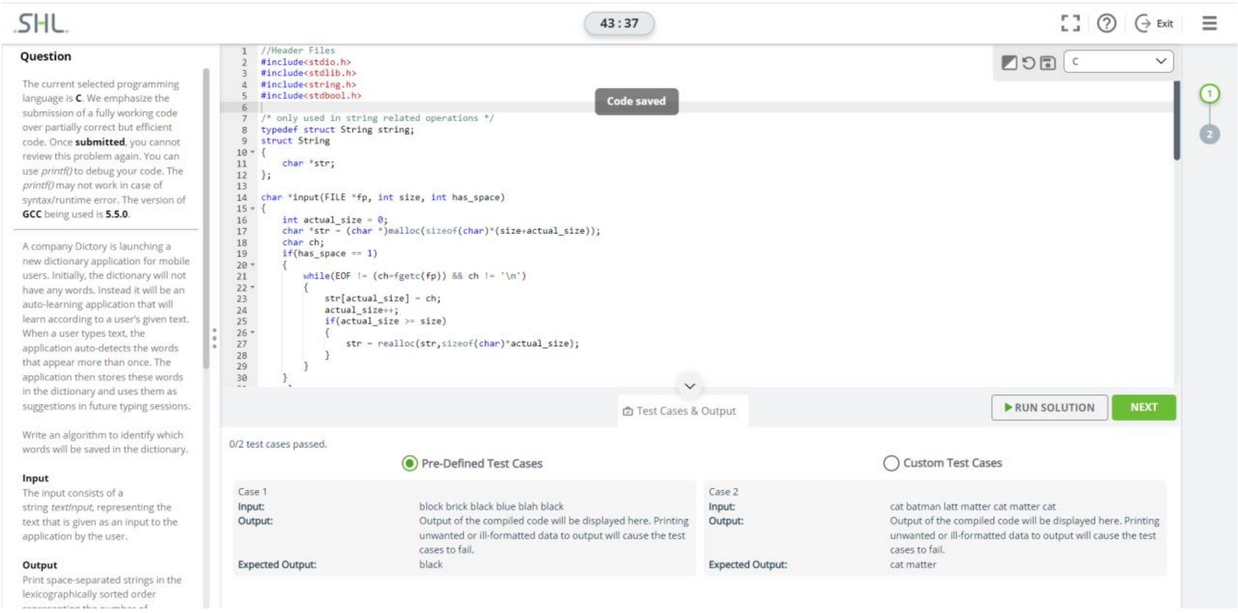
Output

Print space-separated strings in the lexicographically sorted order

```
//Header Files
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>

/* only used in string related operations */
typedef struct String string;
struct String
{
    char *str;
};

char *input(FILE *fp, int size, int has_space)
{
    int actual_size = 0;
    char *str = (char *)malloc(sizeof(char)*(size+actual_size));
    char ch;
    if(has_space == 1)
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
                str = realloc(str,sizeof(char)*actual_size);
            }
        }
    }
    else
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n' && ch != ' ')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
                str = realloc(str,sizeof(char)*actual_size);
            }
        }
    }
    actual_size++;
    str = realloc(str,sizeof(char)*actual_size);
    str[actual_size-1] = '\0';
    return str;
```



Question

The current selected programming language is **C**. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use *printf()* to debug your code. The *printf()* may not work in case of syntax/runtime error. The version of **GCC** being used is **5.5.0**.

A company Dictory is launching a new dictionary application for mobile users. Initially, the dictionary will not have any words. Instead it will be an auto-learning application that will learn according to a user's given text. When a user types text, the application auto-detects the words that appear more than once. The application then stores these words in the dictionary and uses them as suggestions in future typing sessions.

Write an algorithm to identify which words will be saved in the dictionary.

Input

The input consists of a string *textInput*, representing the text that is given as an input to the application by the user.

Output

Print space-separated strings in the lexicographically sorted order

```
//Header Files
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>

/* only used in string related operations */
typedef struct String string;
struct String
{
    char *str;
};

char *input(FILE *fp, int size, int has_space)
{
    int actual_size = 0;
    char *str = (char *)malloc(sizeof(char)*(size+actual_size));
    char ch;
    if(has_space == 1)
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
                str = realloc(str,sizeof(char)*actual_size);
            }
        }
    }
```

0/2 test cases passed.

● Pre-Defined Test Cases ○ Custom Test Cases

Case 1

Input: block brick black blue blah black

Output: Output of the compiled code will be displayed here. Printing unwanted or ill-formatted data to output will cause the test cases to fail.

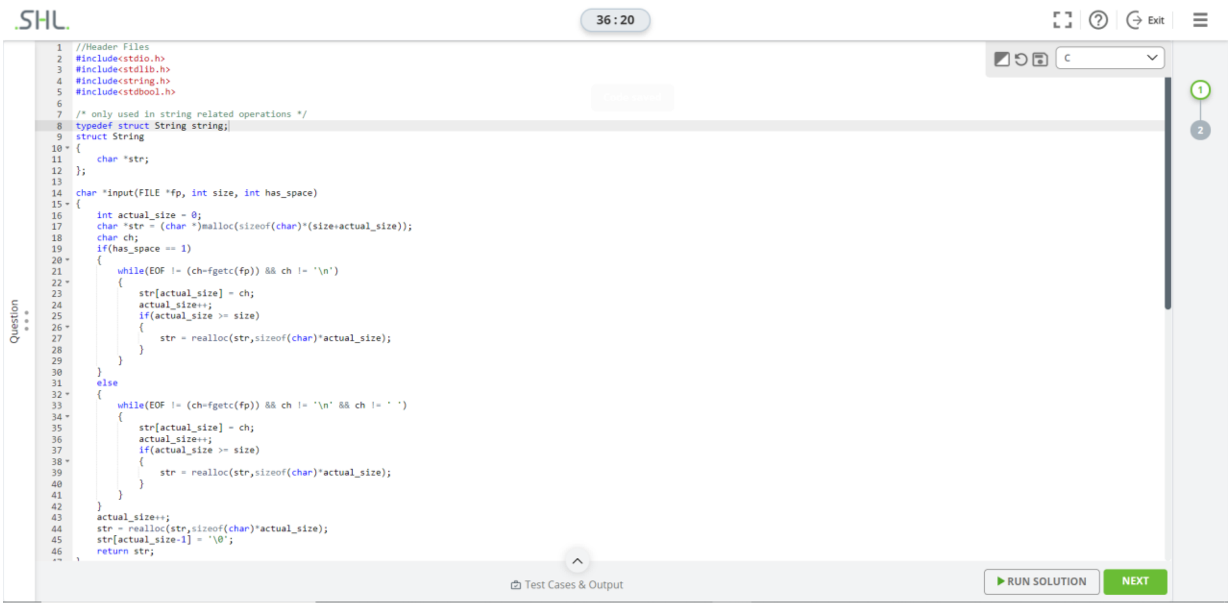
Expected Output: black

Case 2

Input: cat batman latt matter cat matter cat

Output: Output of the compiled code will be displayed here. Printing unwanted or ill-formatted data to output will cause the test cases to fail.

Expected Output: cat matter



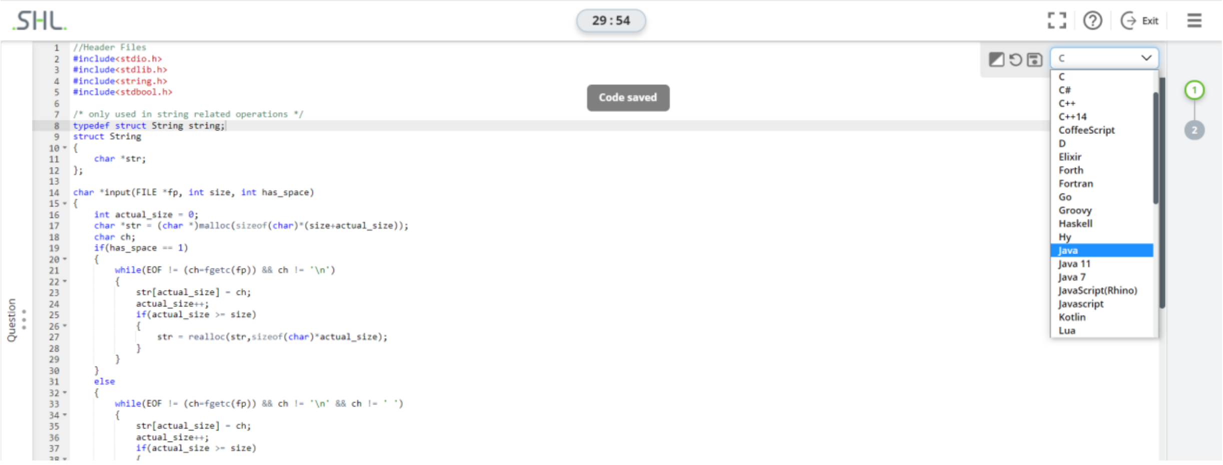
```
//Header Files
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>

/* only used in string related operations */
typedef struct String string;
struct String
{
    char *str;
};

char *input(FILE *fp, int size, int has_space)
{
    int actual_size = 0;
    char *str = (char *)malloc(sizeof(char)*(size+actual_size));
    char ch;
    if(has_space == 1)
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
                str = realloc(str,sizeof(char)*actual_size);
            }
        }
    }
    else
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n' && ch != ' ')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
                str = realloc(str,sizeof(char)*actual_size);
            }
        }
    }
    actual_size++;
    str = realloc(str,sizeof(char)*actual_size);
    str[actual_size-1] = '\0';
    return str;
```

Improvement: Programming languages groups now available now and sorted alphabetically

- Languages in the dropdown menu are sorted alphabetically, and the different versions of a language will appear in groups. This will make it easier to switch language versions while attempting the coding assessment.



```
//Header Files
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>

/* only used in string related operations */
typedef struct String string;
struct String
{
    char *str;
};

char *input(FILE *fp, int size, int has_space)
{
    int actual_size = 0;
    char *str = (char *)malloc(sizeof(char)*(size+actual_size));
    char ch;
    if(has_space == 1)
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
                str = realloc(str,sizeof(char)*actual_size);
            }
        }
    }
    else
    {
        while(EOF != (ch=fgetc(fp)) && ch != '\n' && ch != ' ')
        {
            str[actual_size] = ch;
            actual_size++;
            if(actual_size >= size)
            {
```

Continuous Improvement

Bugs

Minor bugs were fixed as part of our monthly and quarterly maintenance releases.