Testing Documentation

Project Name: Resume Analyzer

Authors: Darragh McGonigle (18316121), Stephen McAleese (18921756)

Supervisor: Michael Scriney

1.Unit Testing

1.1 Server Functions Tests

These are unit tests performed using python’s unittest module to test functions that belong in the ‘server.py’ file of our application.

Test Family			test_gen_skill_colors		
Test Description			Function should take a list of skills and return the list of RGB colours equally spaced around the HSV colour wheel.		
#	Name	Input	Expected Output	Actual Output	Result
1	test_gen_skill_colors_1	['Java', 'Python', 'SQL']	['229,91,91', '91,229,91', '91,91,229']	['229,91,91', '91,229,91', '91,91,229']	Pass
2	test_gen_skill_colors_2	['Java', 'AWS', 'React', 'Node', 'HTML', 'CSS']	['229,91,91', '229,229,91', '91,229,91', '91,229,229', '91,91,229', '229,91,229']	['229,91,91', '229,229,91', '91,229,91', '91,229,229', '91,91,229', '229,91,229']	Pass

1.1.2 Skill Extraction Function

Test Family			test_extract_skills		
Test Description			Function should take text extracted from resume and return a list of technical skills that appear in the document		
#	Name	Input	Expected Output	Actual Output	Result
1	test_extract_skills_1	test_resume1.pdf	['Python', 'JavaScript', 'ReactJS', 'Java']	['Python', 'JavaScript', 'ReactJS', 'Java']	Pass
2	test_extract_skills_2	test_resume2.pdf	['JavaScript', 'ReactJS', 'Java', 'JIRA', 'HTML', 'CSS', 'PostgreSQL', 'Spring']	['JavaScript', 'ReactJS', 'Java', 'JIRA', 'HTML', 'CSS', 'PostgreSQL', 'Spring']	Pass

1.1.4 Calculate Resume Score

Test Family			test_calculate_resume_score		
Test Description			Function should take text extracted from resume, skills and skill occurrence counts and return a dictionary of overall score, length score and skill score		
#	Name	Input	Expected Output	Actual Output	Result
1	test_calculate_resume_score	test_resume1.pdf + [{'name': 'Python'}, {'name': 'JavaScript'}, {'name': 'ReactJS'}, {'name': 'Java'}]	{'overall_score': 40, 'skill_score': 60, 'length_score': 0}	{'overall_score': 40, 'skill_score': 60, 'length_score': 0}	Pass
2	test_calculate_resume_score_2	test_resume2.pdf + [{'name': 'Python'}, {'name': 'JavaScript'}, {'name': 'ReactJS'}, {'name': 'Java'}, {'name': 'AWS'}]	{'overall_score': 48, 'skill_score': 71, 'length_score': 3}	{'overall_score': 48, 'skill_score': 71, 'length_score': 3}	Pass

1.2 Identifier Functions Tests

These are unit tests performed using python’s unittest module to test functions that belong in the ‘identifier.py’ file of our application.

1.2.1 Extract Requirements Function

Test Family			test_extract_requirements		
Test Description			Function to extract skills from text given a body of text and a list of skills.		
#	Name	Input	Expected Output	Actual Output	Result
1	test_extract_requirements	test_data.json[0][description] + skills.csv	["Microservices", "Java", "Go", "Python", "Docker", "Kubernetes", "DevOps", "Jenkins"]	["Microservices", "Java", "Go", "Python", "Docker", "Kubernetes", "DevOps", "Jenkins"]	Pass
2	test_extract_requirements	test_data.json[1][description] + skills.csv	["AWS", "Ruby on Rails", "Java", "MySQL", "Redis", "Terraform", "Docker", "Kafka", "RabbitMQ", "Ruby", "Elasticsearch", "Git", "Microservices"]	["AWS", "Ruby on Rails", "Java", "MySQL", "Redis", "Terraform", "Docker", "Kafka", "RabbitMQ", "Ruby", "Elasticsearch", "Git", "Microservices"]	Pass
3	test_extract_requirements	test_data.json[2][description] + skills.csv	["HTML", "CSS", "JavaScript", "Typescript", "ReactJS", "PHP", "MySQL"]	["HTML", "CSS", "JavaScript", "Typescript", "ReactJS", "PHP", "MySQL"]	Pass
4	test_extract_requirements	test_data.json[3][description] + skills.csv	["Java", "C#", "Go", "Python", "Ruby", "C", "C++", "Rust", "MySQL", "Maven", "JavaScript",	["Java", "C#", "Go", "Python", "Ruby", "C", "C++", "Rust", "MySQL", "Maven", "JavaScript", "NodeJS",	Pass

			"NodeJS", "SQL", "Git", "Hadoop", "HBase", "Unix", "REST", "Scrum", "Posix", "AWS"]	"SQL", "Git", "Hadoop", "HBase", "Unix", "REST", "Scrum", "Posix", "AWS"]	
5	test_extract_requirements	test_data.json[4][description] + skills.csv	["Java", "Kotlin", "ReactJS", "GraphQL", "Storybook", "PostgreSQL", "DynamoDB", "RabbitMQ", "R", "TailwindCSS"]	["Java", "Kotlin", "ReactJS", "GraphQL", "Storybook", "PostgreSQL", "DynamoDB", "RabbitMQ", "R", "TailwindCSS"]	Pass

1.2.2 Text Normalisation Function

Test Family			test_normalize_text		
Test Description			Function to remove any non alphanumeric characters from a description..		
#	Name	Input	Expected Output	Actual Output	Result
1	test_normalize_text	test_data.json[0][description]	All Characters AlphaNumeric = True	All Characters AlphaNumeric = True	Pass
2	test_normalize_text	test_data.json[1][description]	All Characters AlphaNumeric = True	All Characters AlphaNumeric = True]	Pass
3	test_normalize_text	test_data.json[2][description]	All Characters AlphaNumeric = True	All Characters AlphaNumeric = True	Pass
4	test_normalize_text	test_data.json[3][description]	All Characters AlphaNumeric = True	All Characters AlphaNumeric = True	Pass
5	test_normalize_text	test_data.json[4][description]	All Characters AlphaNumeric = True	All Characters AlphaNumeric = True	Pass

1.2.3 Years of Experience Extraction Function

Test Family			test_get_years_of_experience		
Test Description			Function to extract the years of experience required for a job post		
#	Name	Input	Expected Output	Actual Output	Result
1	test_get_years_of_experience	test_data.json[0][description]	[2]	[2]	Pass
2	test_get_years_of_experience	test_data.json[1][description]	[ ]	[]	Pass
3	test_get_years_of_experience	test_data.json[2][description]	[ ]	[]	Pass
4	test_get_years_of_experience	test_data.json[3][description]	[5,1,4,2,2]	[5, 1 ,4 ,2 ,2]	Pass
5	test_get_years_of_experience	test_data.json[4][description]	[2]	[2]	Pass

1.2.4 Vectorize Text Function

Test Family			test_vectorize_text		
Test Description			Function to convert text into 17 element vector using LDA model		
#	Name	Input	Expected Output	Actual Output	Result
1	test_vectorize_text	test_data.json[0][description]	len(vec) = 17	len(vec) = 17	Pass
2	test_vectorize_text	test_data.json[1][description]	len(vec) = 17	len(vec) = 17	Pass
3	test_vectorize_text	test_data.json[2][description]	len(vec) = 17	len(vec) = 17	Pass
4	test_vectorize_text	test_data.json[3][description]	len(vec) = 17	len(vec) = 17	Pass
5	test_vectorize_text	test_data.json[4][description]	len(vec) = 17	len(vec) = 17	Pass

1.2.5 Get LDA Results Function

Test Family			test_get_lda		
Test Description			Wrapper function that uses previous ‘vectorize_text’ to create a tuple of job id, job desc and lda vec’		
#	Name	Input	Expected Output	Actual Output	Result
1	test_get_lda	1, test_data.json[0][description]	result[0] =1 result[1] = test_data.json[0][description] len(result[2]) = 17	result[0] =1 result[1] = test_data.json[0][description] len(result[2]) = 17	Pass
2	test_get_lda	2, test_data.json[1][description]	result[0] =2 result[1] = test_data.json[1][description] len(result[2]) = 17	result[0] =2 result[1] = test_data.json[1][description] len(result[2]) = 17	Pass
3	test_get_lda	3, test_data.json[2][description]	lresult[0] =3 result[1] = test_data.json[2][description] len(result[2]) = 17	lresult[0] =3 result[1] = test_data.json[2][description] len(result[2]) = 17	Pass
4	test_get_lda	4, test_data.json[3][description]	result[0] =4 result[1] = test_data.json[3][description] len(result[2]) = 17	result[0] =4 result[1] = test_data.json[3][description] len(result[2]) = 17	Pass
5	test_get_lda	5, test_data.json[4][description]	result[0] =5 result[1] = test_data.json[4][description] len(result[2]) = 17	result[0] =5 result[1] = test_data.json[4][description] len(result[2]) = 17	Pass

1.2.5 Rule Generation Function

Test Family			test_get_rules		
Test Description			Function to generate rules from translation list using apriori's algorithm		
#	Name	Input	Expected Output	Actual Output	Result
1	test_get_rules	[["Java", "Python", 'Spring'], ["Java", "Python"], ["CSS", "HTML"], ["CSS", "HTML", "ReactJS"]]	rules[0] = [['HTML'], 'CSS', 0.5, 1.0, 2.0] **Or** [['CSS'], 'HTML', 0.5, 1.0, 2.0]	rules[0] = [['HTML'], 'CSS', 0.5, 1.0, 2.0]	Pass

2.Integration Testing

2.1 Backend API Tests

These test were created using the Postman API testing tool and were then bundled into a collection which could be run using the Newman Postman runner

2.1.1 Status Check

Request Name			Status_Check		
Request Method			GET		
Request Description			Request to get a simple response string to indicate the server is running.		
#	Name	Input	Expected Output	Actual Output	Result
1	Response Code		200	200	Pass
2	Status Message		“Server is running”	“Server is running”	Pass

2.1.2 Resume Upload

Request Name			Resume_Upload_1		
Request Method			POST		
Request Description			Request to upload resume for analysis to the server		
#	Name	Input	Expected Output	Actual Output	Result
1	Status Code	test-resume.pdf	200	200	Pass
2	Skills	test-resume.pdf	[ReactJS, Python, Java, JavaScript]	[ReactJS, Python, Java, JavaScript]	Pass
3	Recommendations	test-resume.pdf	recommendations.length = 10	recommendations.length = 10	Pass
4	Skill Count	test-resume.pdf	skill_count[‘.NET’] = 78	skill_count[‘.NET’] = 78	Pass
5	Jobs	test-resume.pdf	Jobs.length > 0	Jobs.length > 0	Pass
6	Resume_Score	test-resume.pdf	Overall_score = 40 Skill_score = 60 Length_score = 0	Overall_score = 40 Skill_score = 60 Length_score = 0	Pass

Request Name			Resume_Upload_2		
Request Method			POST		
Request Description			Request to upload resume for analysis to the server		
#	Name	Input	Expected Output	Actual Output	Result
1	Status Code	test-resume.pdf	200	200	Pass
2	Skills	test-resume.pdf	[HTML, CSS, JAVA, ReactJS, JIRA, JavaScript, PostgreSQL, Spring]	[HTML, CSS, JAVA, ReactJS, JIRA, JavaScript, PostgreSQL, Spring]	Pass
3	Recommendations	test-resume.pdf	recommendations.length = 10	recommendations.length = 10	Pass

4	Skill Count	test-resume.pdf	skill_count[‘.NET’] = 78	skill_count[‘.NET’] = 78	Pass
5	Jobs	test-resume.pdf	Jobs.length > 0	Jobs.length > 0	Pass
6	Resume_Score	test-resume.pdf	Overall_score = 43 Skill_score = 63 Length_score = 3	Overall_score = 43 Skill_score = 63 Length_score = 3	Pass

2.1.3 Path Data

Request Name			Path_Data		
Request Method			GET		
Request Description			Request to get skill counts broken down by role type		
#	Name	Input	Expected Output	Actual Output	Result
1	Response Code		200	200	Pass
2	Response Body		Contains 18 Roles = True (See footnote)	True	Pass

Footnote:

18 Roles are as follows:
'Junior Frontend Developer', 'Senior Frontend Developer', 'Junior Backend Developer', 'Senior Backend Developer', 'Junior Full Stack Developer', 'Full Stack Developer', 'Senior Full Stack Developer', 'QA Engineer', 'Senior QA Engineer', 'Business Analyst', 'Development Lead', 'Software Architect', 'Product Owner', 'Project Manager', 'Devops', 'Senior Devops', 'Automation Engineer', 'Cloud Engineer', 'Database Admin (DBA)'

2.1.4 Role Data

Request Name			Role_Data_Junior_FE		
Request Method			GET		
Request Description			Request to get a job list based on input string		
#	Name	Input	Expected Output	Actual Output	Result
1	Response Code	Junior Frontend Developer	200	200	Pass
2	Response Body	Junior Frontend Developer	jobs.to.exist = True	True	Pass

Request Name			Role_Data_Devops		
Request Method			GET		
Request Description			Request to get a job list based on input string		
#	Name	Input	Expected Output	Actual Output	Result
1	Response Code	Devops	200	200	Pass
2	Response Body	Devops	jobs.to.exist = True	True	Pass

2.1.4 Report Data

Request Name			Role_Data_Junior_FE		
Request Method			GET		
Request Description			Get all table data for reports page		
#	Name	Input	Expected Output	Actual Output	Result
1	Response Code		200	200	Pass
2	Skill_Counts		skill_counts.to.exist=True	True	Pass
3	Soft_Skills_Counts		Soft_skill_counts.to.exist = True	True	Pass
4	Years_of_Experience		Years_of_experience.to.exist = True	True	Pass
5	Locations		Location_counts = True	True	Pass

2.2 Frontend Integration Tests

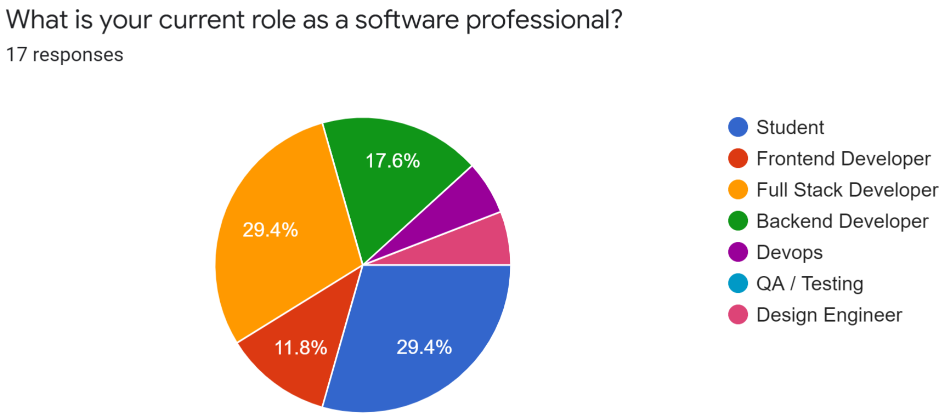
Frontend integration tests created with the Jest testing library to ensure pages load properly and data is correctly displayed

Test Name			Frontend Upload test		
Test Description			Mimic user accessing page and uploading resume then browsing each section		
#	Action	Input	Expected Output	Actual Output	Result
1	User loads App		Upload page rendered	Upload page rendered	Pass
2	Resume Upload	testResume.pdf	Submit button appears	Submit button appears	Pass
3	Submit Resume	Click on submit	Results page rendered	Results page rendered	Pass
4	Navigate to Reports	Click on Reports Icon	Reports page rendered	Reports page rendered	Pass

3.User Testing

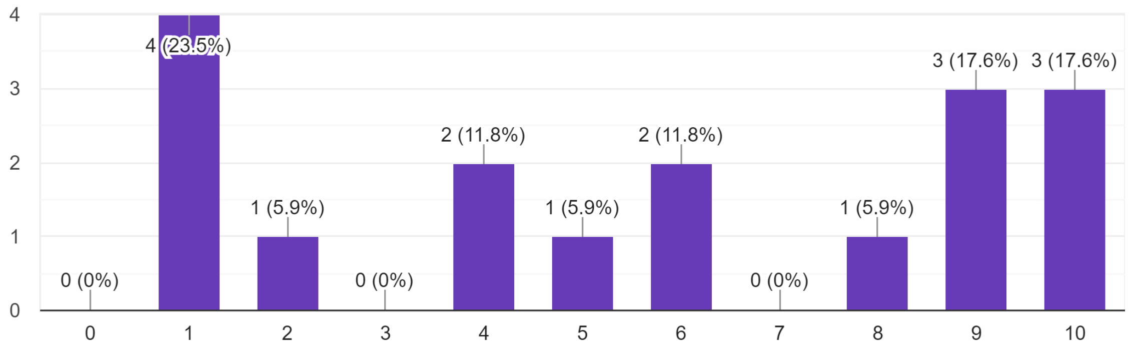
Feedback Form: https://forms.gle/WVertvJDmwxtcJLN6
Testing Results: Results

The first section each tester fills out some non-identifying background information (Profession, Experience, Opinions on Industry). This gives us background into who the testers are and what they want out of the job finding market. See results of the first section below.


What is your current role as a software professional?
17 responses

How many years of experience do you have?

17 responses



What is your current process for finding software jobs?

16 responses

LinkedIn

through reference or through job portals, words of mouth

Freelance Developer

You have a colleague or friend in the company who suggested you apply or mentioned they were hiring

Applying Online on Job Portals

Looking to switch job

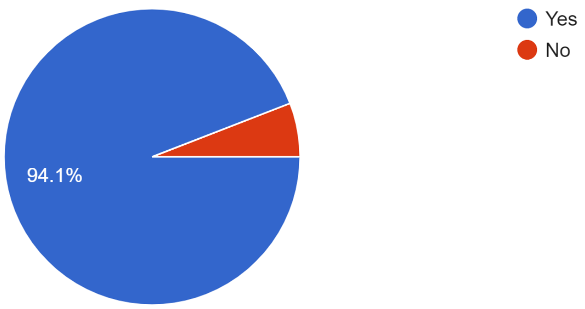
Job websites

Indeed, LinkedIn

I use LinkedIn to find jobs and build contacts.

Do you think the job finding experience could be improved for software professionals?

17 responses

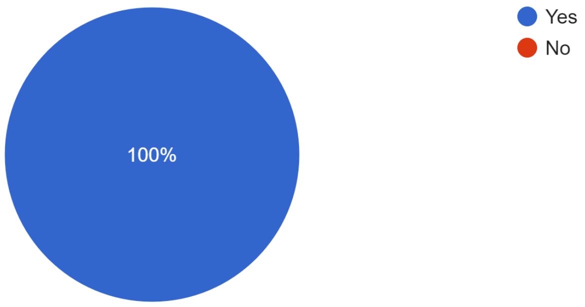


In the next section the user was given 4 tasks to complete on the web application

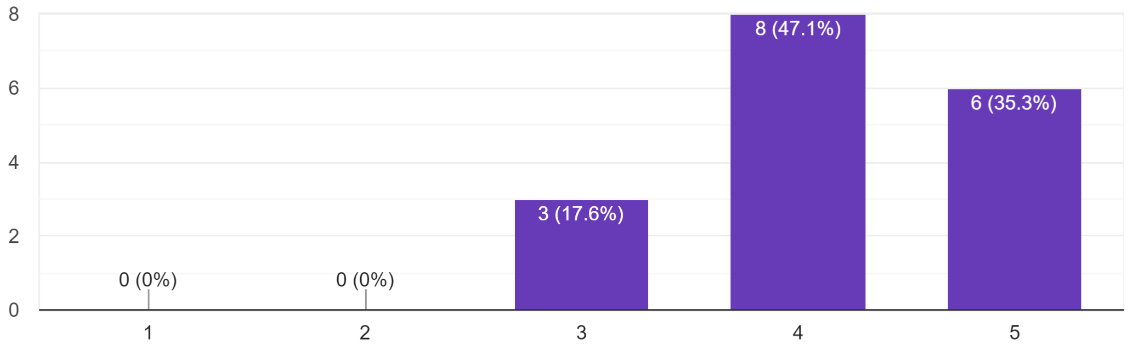
1. Upload a Resume
2. Explore the Dashboard
3. Explore the Reports Section
4. Explore the Career Path Section

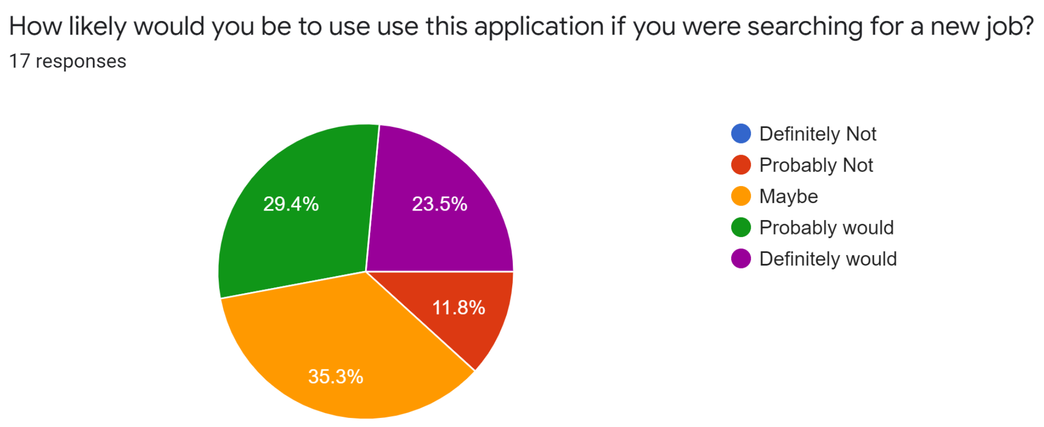
Once complete the user was then asked questions about their testing experience and to give feedback. See feedback below.

Were you able to complete all the tasks on your own?
17 responses



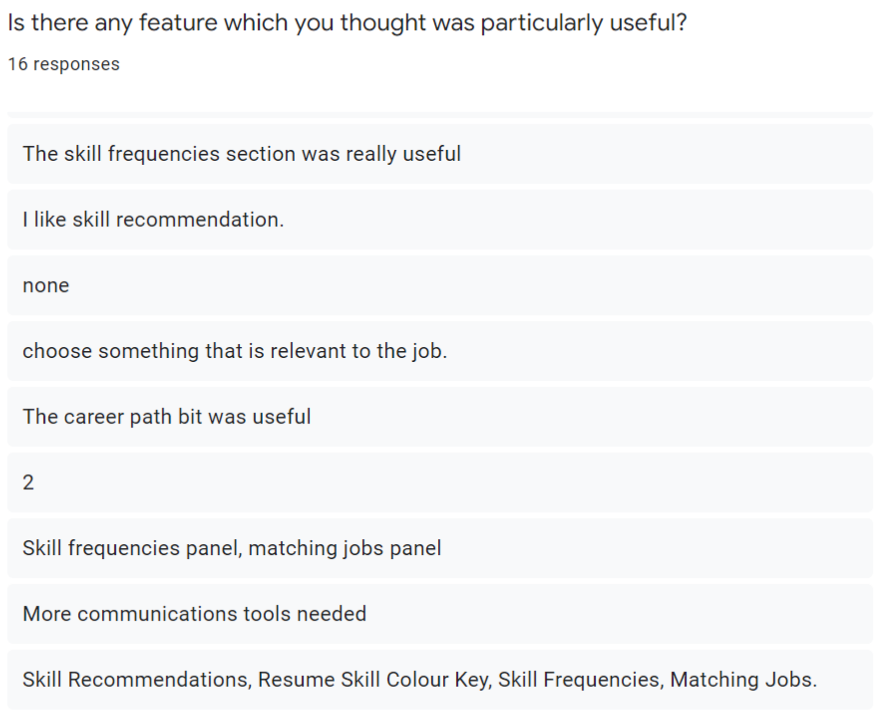
On a scale from 1 to 5 how useful do you think the app is?
17 responses





How likely would you be to use use this application if you were searching for a new job?

17 responses



Is there any feature which you thought was particularly useful?

16 responses

The skill frequencies section was really useful

I like skill recommendation.

none

choose something that is relevant to the job.

The career path bit was useful

2

Skill frequencies panel, matching jobs panel

More communications tools needed

Skill Recommendations, Resume Skill Colour Key, Skill Frequencies, Matching Jobs.

Is there any features you would like to see added?

16 responses

none

Talk about skills you're currently developing.

The ability to save the resume so I don't have to reupload it every time

5

Break the resume score down by skill path, allow the user to choose a location, show how old the job posts are, it might be useful if you could convert the resume analysis into a PDF report

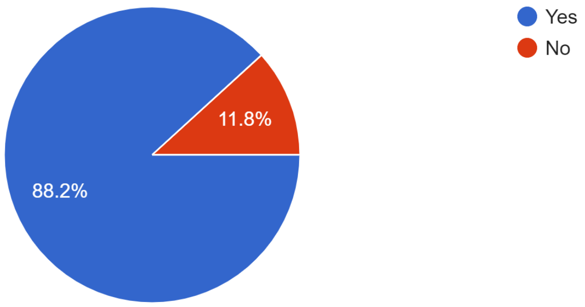
More styling

Courses/ Certificate Recommendations to progress down outlines career paths.

Easier job finding

Were you able to read all text on the site easily?

17 responses



Did you encounter any bugs / problems / issues during your testing

16 responses

Web site does not working, pls chk.

Unstable Environment. ...

No, not faced any issue

Nope

Nothing

The resume could be analyzed faster

Draggable sections, Non standardized font/ sizing, tree progress meter not working as intended, dots in tree need to be spaced, length did not measure correctly leading to incorrect Resume score.

It took a while to load

Is there any changes you would like to see made to the site?

15 responses

All good at present but if added eye catching report names it will be useful

The User interface can be enhanced in much colorful and intuitive way

The fonts are a bit inconsistant

More career options

Add more analytics

Nothing

Styling

Tree flow in opposite direction, use more relevant information in reporting section. When hovering over tabs in left navigation column it should highlight what page you're on.

Any other feedback?

9 responses

no.

good

Make the charts more colourful

Analysis was fast.

none

Nice

yes

Thanks for your great outstanding survey and i would thank you for giving me the chance to participate and you are helping make me better.

The layout of the reporting page is confusing.