

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the text 'CST 2120'.

CST 2120

Project Report

Hamza Bari M00668318

Several thin, dark blue wavy lines originate from the bottom left corner and curve upwards and to the right, creating a decorative element.

Contents

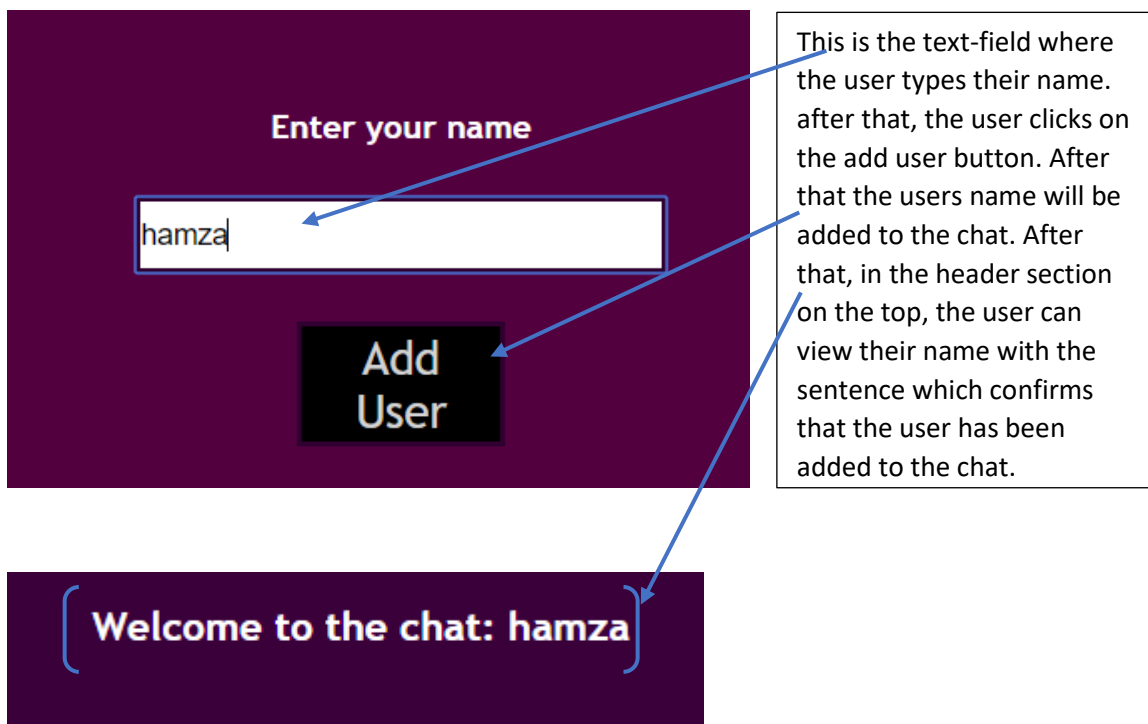
Introduction	2
Description of the functionality of the Website.....	2
Database Design	7
Screenshots of the Webpage.....	7
Testing	14
HTML & CSS Validation Testing	14
HTML Validation	14
CSS Validation	15
Selenium Tests	16
Video Reference	20

Introduction

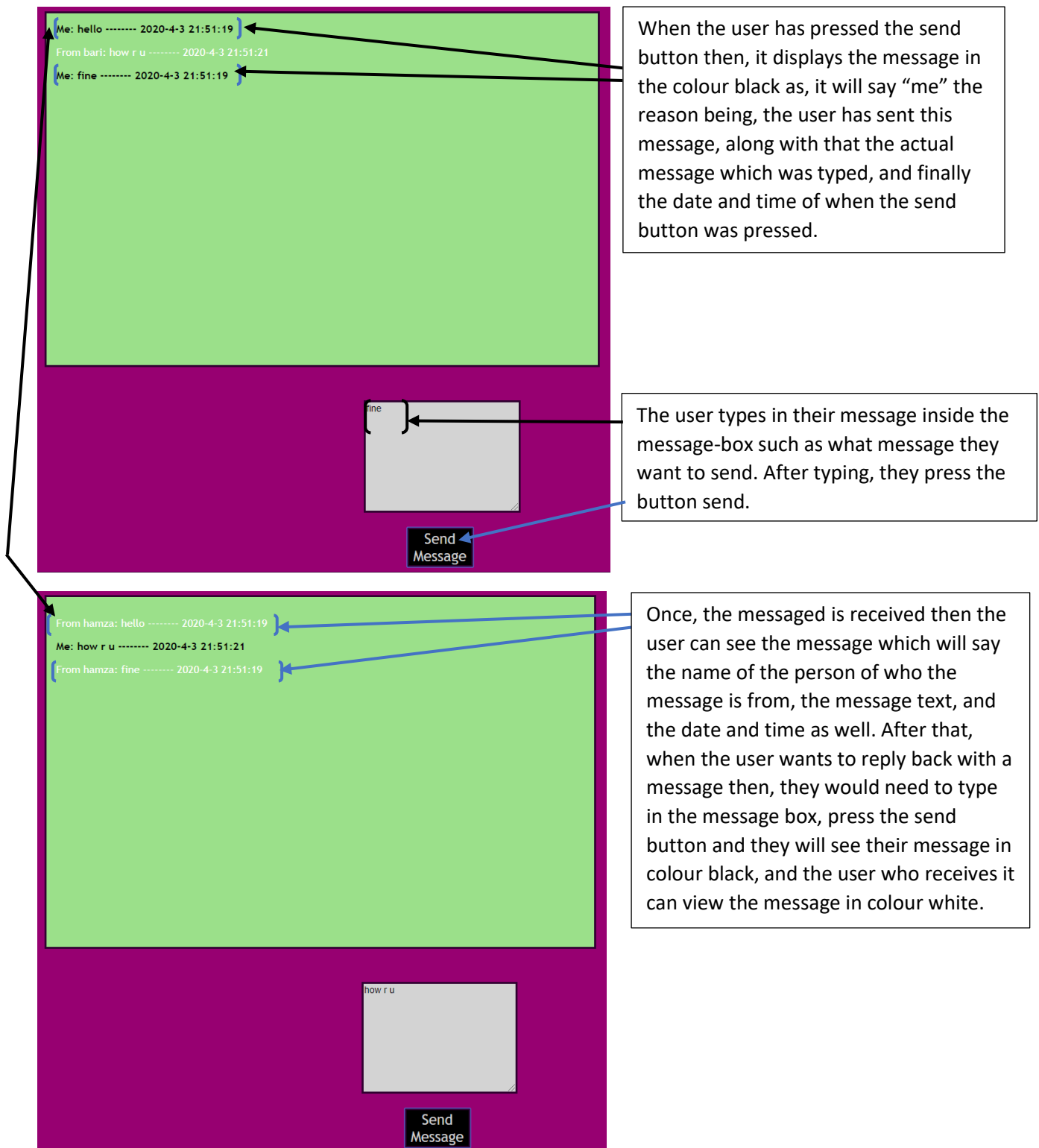
In this introduction I'm going to talk about what this document contains: I have explained the description of the functionality of the website by using screenshots and annotations, I have also provided the database diagram of the database is working also, in terms of storage and retrieval, I have also provided different webpage outputs of screenshots which includes the key functionality, and I've documented my HTML & CSS validations tests as well as the website automated browser testing by using selenium. I have also provided the YouTube video demonstration link. The rest of the information can be viewed in the sections below.

Description of the functionality of the Website

Below, I'm going to use screenshots and annotation to explain the functionality of the website in detail. This includes the following: how the messages are sent and received between the users, how data is stored in the database, how data is retrieved from the database, and finally how the webpage changes with some of the button interaction. To describe briefly my project is a chat application where multiple users can add themselves to the chat and send messages to each other, and users will send and receive each other message. The users can view the chat history in the table format to see the previous messages sent and received. The user can also click on certain buttons change the webpage, and the users can give feedback about the website and chat in the feedback section.



Just like I've shown above, how a user can add themselves to the chat, this way many users can add themselves to the chat. For example, 10 users add themselves to the chat, and when 1 user sends the message all the other 9 users receive the message in white colour which indicates that they have received this message from 1 user but, the user who has sent the message will see the message is well in colour black which indicates that the user themselves have sent this message. In my example below, I'm going to show 2 users.



The username with the generated id will get stored inside the users SQL database table. The message table will have a foreign key of the user id from the users table and including the foreign key the table is going to contain the actual text messages and the date and time for that message. So, next to the user id will be the message which the user has sent as well as the date and time.

In addition to that, when sending the messages, and messages being added to the chat-box and being received from the chat-box can also be considered as an example of the webpage changing on the button interactions.

chatapplication.users: 2 rows total (approximately)

id	username
793	hamza
123	bari

This is the user id with the username stored in the database. The user id is the primary key.

chatapplication.messages: 3 rows total (approximately)

userID	messages	datetime
123	how r u	2020-4-3 15:27:41
123	fine	2020-4-3 15:27:41
793	hi	2020-4-3 20:43:55

This is the messages table where it's using the foreign key user id, and it shows what message the user has sent, as well as the data and time of the message in terms of when it was sent. Note: you can see the annotation above.

Chat History

Look below to see all the history of the messages.

User ID	Messages	Date & Time
123	how r u	2020-4-3 15:27:41
123	fine	2020-4-3 15:27:41
793	hi	2020-4-3 20:43:55
419	h	2020-4-3 23:45:46
419	h	2020-4-3 23:45:46

The screenshot above shows the chat-history section of the webpage where it loads up the data from the messages table and it displays the data in the HTML table. This is where the users can see the previous messages, they have sent over this chat application. For example, if the users want to see what message they have just sent they can come down to this section and view the message, and also if the user has missed the message they can also come down to this table and view the message. This can also be considered as an example of webpage being changed.

Welcome to the chat: bari

Feedback

When the user clicks on this feedback button on the top on the webpage header then, it displays the extra HTML content by the click of this button. This is one example of the webpage changing on the button clicking.

Give Feedback

This is where you can upload your text file to give us feedback or comments.

Enter your feedback here:

Send

ok

Send

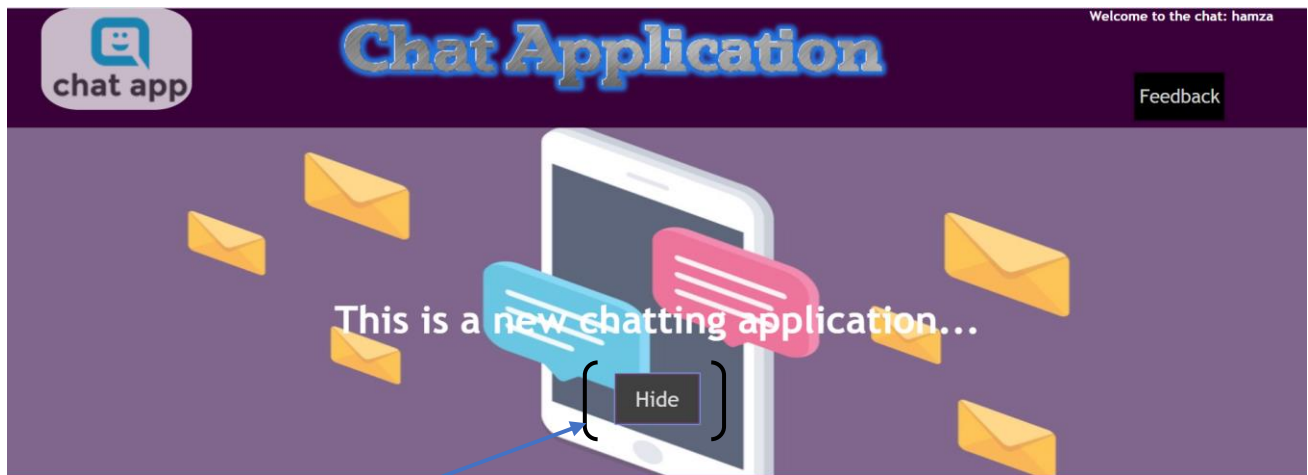
Sent

The user can type in this text-area when they want to give feedback about the website. After that, the user presses the send button to send the feedback message and they receive a response **“Sent”** underneath the send button. The feedback text will eventually get stored in the database but, the storage of the feedback message is partially working and it send the feedback message data to the database after the new users have been added to the chat and have sent and received messages.

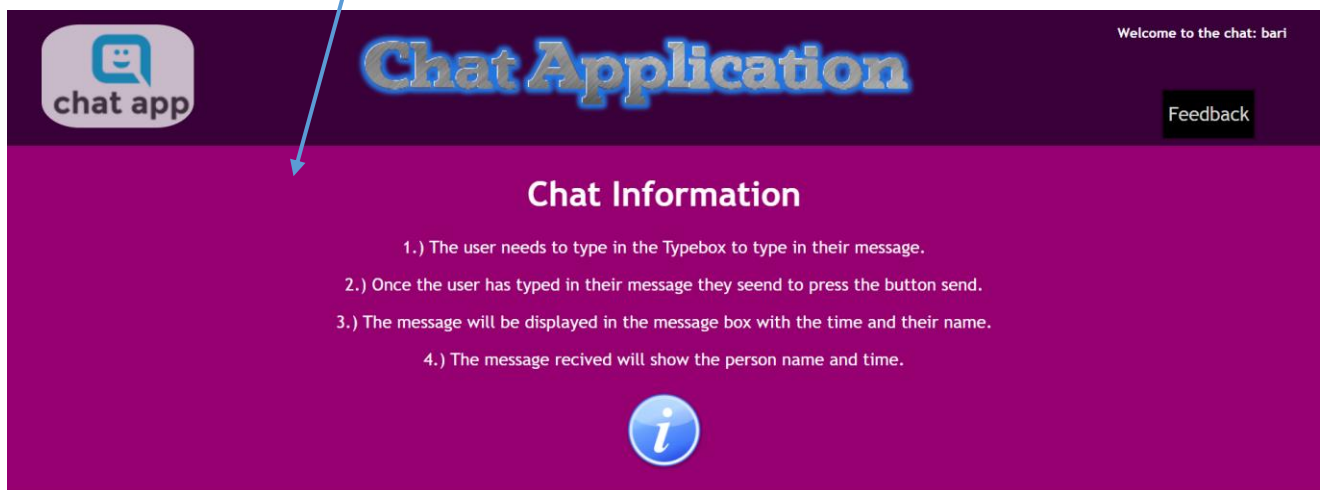
chatapplication.feedback: 1 rows total (approximately)

userResponse
ok

This is the feedback table, where all the feedback messages are going to be stored inside.



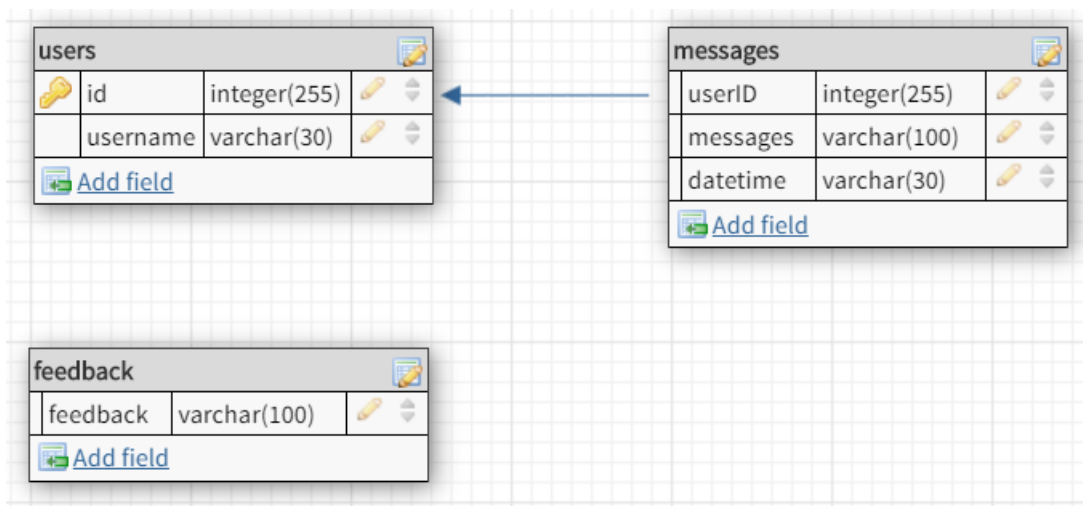
When the user clicks on this hide button then, it removes the header image from the webpage. As, you can see that from the example below.



This can be considered as an example of the webpage changing on the button interaction.


Database Design

Below, is the diagram of my database design. This is how the webpage data is being stored into the database, and this is also how data is being retrieved from the database to the webpage.




Screenshots of the Webpage

There are three different screenshots of the webpage which display the different outputs of the webpage as, to when the button has been clicked. The first screenshot shows when the webpage is loaded up on the browser for the first time. The second screenshot shows when you send and receive the messages it's displayed in the chat-box which is known, as the main functionality of the website and finally the third screenshot shows different outputs and interactions of the webpage when some of the buttons have been clicked. On the pages below you can view the screenshots.



Chat Application




This is a new chatting application...

Hide

Chat Information

- 1.) The user needs to type in the Typebox to type in their message.
- 2.) Once the user has typed in their message they send to press the button send.
- 3.) The message will be displayed in the message box with the time and their name.
- 4.) The message received will show the person name and time.




Join the Chat Information:

- 1.) Enter your name into the textbox.
- 2.) Press the button "add" to be added to the chat.
- 3.) If you are successful then, your name will be displayed on the top-right hand corner.
- 4.) Enjoy!


Enter your name

Please Enter your name here:

Add User



Chat Messages Area




Enter your message here:


Send Message

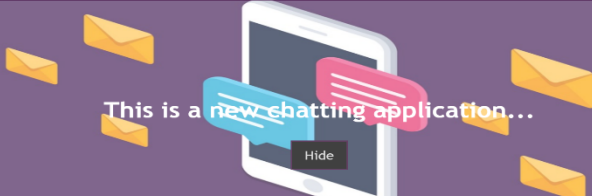
Chat History

Look below to see all the history of the messages.

© 2020 | Hamza Bari




Chat Application
Welcome to the chat: hamza
Feedback




This is a new chatting application...

Hide

Chat Information

- 1.) The user needs to type in the Typebox to type in their message.
- 2.) Once the user has typed in their message they send to press the button send.
- 3.) The message will be displayed in the message box with the time and their name.
- 4.) The message received will show the person name and time.




Join the Chat Information:

- 1.) Enter your name into the textbox.
- 2.) Press the button "add" to be added to the chat.
- 3.) If you are successful then, your name will be displayed on the top-right hand corner.
- 4.) Enjoy!

Enter your name

Please Enter your name here:

Add User


Chat Messages Area

Me: hello ----- 2020-4-2 2:43:3

From bari: how r u ----- 2020-4-2 2:41:27

Send Message

Chat History

Look below to see all the history of the messages.

User ID	Messages	Date & Time
787	hello	2020-4-2 2:41:3
345	how r u	2020-4-2 2:41:27
787	i am good	2020-4-2 2:41:3
345	you	2020-4-2 2:41:27
787	perfectly fine	2020-4-2 2:41:3
345	ok	2020-4-2 2:41:27
705	hello	2020-4-2 2:43:3

Give Feedback


This is where you can upload your text file to give us feedback or comments.


its is a good website

Send

Sent

© 2020 | Hamza Bari

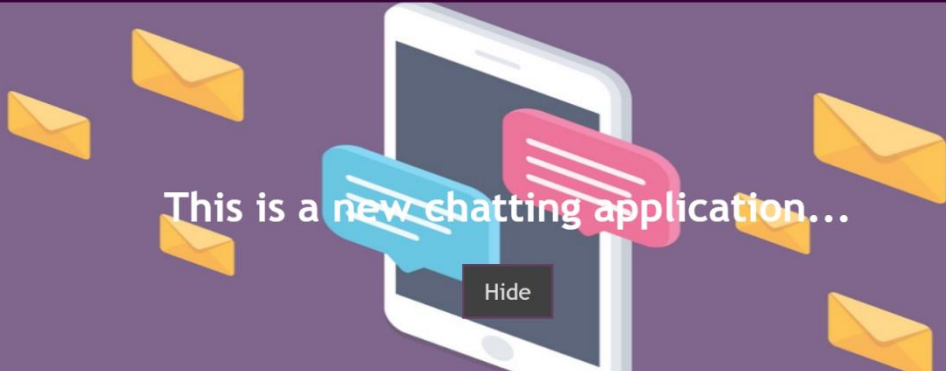




Chat Application

Welcome to the chat: bari

Feedback




This is a new chatting application...

Hide

Chat Information

- 1.) The user needs to type in the Typebox to type in their message.
- 2.) Once the user has typed in their message they send to press the button send.
- 3.) The message will be displayed in the message box with the time and their name.
- 4.) The message received will show the person name and time.




Join the Chat Information:

- 1.) Enter your name into the textbox.
- 2.) Press the button "add" to be added to the chat.
- 3.) If you are successful then, your name will be displayed on the top-right hand corner.
- 4.) Enjoy!

Enter your name

Please Enter your name here:

Add User



Chat Messages Area

Me: how r u ----- 2020-4-2 2:41:27
From hamza: i am good ----- 2020-4-2 2:41:3
Me: you ----- 2020-4-2 2:41:27
From hamza: perfectly fine ----- 2020-4-2 2:41:3
Me: ok ----- 2020-4-2 2:41:27
From hamza: hello ----- 2020-4-2 2:43:3
Me: how r u ----- 2020-4-2 2:41:27

how r u

Send
Message

Chat History

Look below to see all the history of the messages.

User ID	Messages	Date & Time
787	hello	2020-4-2 2:41:3
345	how r u	2020-4-2 2:41:27
787	i am good	2020-4-2 2:41:3
345	you	2020-4-2 2:41:27
787	perfectly fine	2020-4-2 2:41:3
345	ok	2020-4-2 2:41:27

Give Feedback

This is where you can upload your text file to give us feedback or comments.


this is a good website

Send

Sent


© 2020 | Hamza Bari

[f](#) [t](#) [g+](#) [p](#)


Chat Application
Welcome to the chat: hamza
Feedback

Chat Information

- 1.) The user needs to type in the Typebox to type in their message.
- 2.) Once the user has typed in their message they send to press the button send.
- 3.) The message will be displayed in the message box with the time and their name.
- 4.) The message received will show the person name and time.




Join the Chat Information:

- 1.) Enter your name into the textbox.
- 2.) Press the button "add" to be added to the chat.
- 3.) If you are successful then, your name will be displayed on the top-right hand corner.
- 4.) Enjoy!

Enter your name

Please Enter your name here.

Add User



Chat Messages Area

Hi: hi ----- 2020-4-3 20:43:55

From Bari: how r u ----- 2020-4-3 20:43:57

hi

Send Message

Chat History

Look below to see all the history of the messages.

User ID	Messages	Date & Time
373	hi	2020-4-3 15:27:35
123	how r u	2020-4-3 15:27:41
373	k	2020-4-3 15:27:35
123	fine	2020-4-3 15:27:41
793	hi	2020-4-3 20:43:55


Give Feedback

This is where you can upload your text file to give us feedback or comments.

Enter your feedback here:

Send

© 2020 | Hamza Bari



Testing

This is where I will be documenting my tests which include the following: HTML & CSS validation, and the selenium tests.

HTML & CSS Validation Testing

HTML Validation

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file index.html

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Error** The `center` element is obsolete. [Use CSS instead.](#)

From line 22, column 13; to line 22, column 20

`<center><button`

2. **Error** The `center` element is obsolete. [Use CSS instead.](#)

From line 34, column 9; to line 34, column 16

`<center><img s`

3. **Error** Bad value `_blank_` for attribute `target` on element `a`: Reserved keyword `blank_` used.

From line 72, column 13; to line 72, column 64

` <img`

This is when I tested the index html webpage for the first attempt. As, you notice I'm getting errors.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for index.html

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.


Used the HTML parser.

Total execution time 22 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 20.3.23

As, you can now see on my second attempt the index HTML webpage validation has passed on the second attempt as, I've fixed all the HTML errors.

CSS Validation

 The W3C CSS Validation Service
W3C CSS Validator results for styles.css (CSS level 3 + SVG)

Jump to: [Errors \(1\)](#) [Validated CSS](#)

W3C CSS Validator results for styles.css (CSS level 3 + SVG)


Sorry! We found the following errors (1)

URI : styles.css

61	#logo2	Value Error : float <code>center</code> is not a float value : <code>center</code>
----	--------	--

↑ TOP

This is when I tested the CSS stylesheet for the first attempt. As, you notice I'm getting an error.

 The W3C CSS Validation Service
W3C CSS Validator results for styles.css (CSS level 3 + SVG)

Jump to: [Validated CSS](#)

W3C CSS Validator results for styles.css (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as [CSS level 3 + SVG](#) !

As, you can now see on my second attempt the CSS Stylesheet validation has passed on the second attempt as, I've fixed all the CSS errors.

Selenium Tests

Project: TestForName*

Search tests...

http://localhost:8080/

	Command	Target	Value
1	open	http://localhost:8080/	
2	set window size	1552x840	
3	click	id=fname	
4	type	id=fname	test-hamza
5	click	id=btnAddUser	
6	assert text	css=b:nth-child(2)	test-hamza

Command:

Target:

Value:

Description:

Log	Reference
1. open on http://localhost:8080/ OK	22:52:52
2. setWindowSize on 1552x840 OK	22:52:52
3. click on id=fname OK	22:52:52
4. type on id=fname with value test-hamza OK	22:52:53
5. click on id=btnAddUser OK	22:52:53
6. assertText on css=b:nth-child(2) with value test-hamza OK	22:52:53
'Untitled' completed successfully	22:52:53

This test was for testing the input of the name such as, when the user inputs the name in the text field and presses the button add then, it should add the username value on the top-header.

Selenium IDE - TestForName*

Project: TestForName*

Tests ▾ +

Search tests... 🔍

test1

test2*

test3*

http://localhost:8080/

	Command	Target	Value
1	open	http://localhost:8080/	
2	set window size	1552x840	
3	click	id=fname	
4	type	id=fname	hamza
5	click	id=btnAddUser	
6	click	id=text	
7	type	id=text	test, how r u?
8	click	id=btnSend	
9	mouse over	id=btnSend	
10	mouse out	id=btnSend	
11	assert text	css=strong:nth-child(2)	test, how r u?

Command: assert text // [🔗]

Target: css=strong:nth-child(2) [🔗] [🔍]

Value: test, how r u?

Description:

Log Reference

6. click on id=text OK	00:37:10
7. type on id=text with value test, how r u? OK	00:37:10
8. click on id=btnSend OK	00:37:10
9. mouseOver on id=btnSend OK	00:37:11
10. mouseOut on id=btnSend OK	00:37:11
11. assertText on css=strong:nth-child(2) with value test, how r u? OK	00:37:11
'test2' completed successfully	00:37:12

This test was for when the user sends a message then, the message value typed in the chat box is displayed in the message area.

Selenium IDE - TestForName*

Project: TestForName*

Executing ▾

test3*

http://localhost:8080/

	Command	Target	Value
1	open	http://localhost:8080/	
2	set window size	1552x840	
3	click	id=btnHide	
4	assert element not present	css=.header	

Command: assert element not present

Target: css=.header

Value:

Description:

Runs: 1 Failures: 0

Log Reference

'test3' ended with 1 error(s) 00:31:16

Running 'test3' 00:34:57

1. open on http://localhost:8080/ OK 00:34:57

2. setWindowSize on 1552x840 OK 00:34:57

3. click on id=btnHide OK 00:34:58

4. assertElementNotPresent on css=.header OK 00:34:58

'test3' completed successfully 00:34:58

This test is for checking if the image element gets removed from the webpage after clicking the button hide.

Selenium IDE - TestForName*

Project: TestForName*

Tests ▾ + ▶ ▶ ⚙ ⌛

Search tests... 🔍

http://localhost:8080/ ▾

	Command	Target	Value
test1			
test2*	4 type	id=fname	hamza
test3*	5 click	id=btnAddUser	
test4*	6 click	id=text	
	7 type	id=text	hi
	8 click	id=btnSend	
	9 mouse over	id=btnSend	
	10 mouse out	id=btnSend	
	11 click	id=btnFeedback	
	12 assert element present	id=class	

Command: assert element present // [🔗]

Target: id=class [🔍]

Value:

Description:

Log Reference

7. type on id=text with value hi	OK	00:52:39
8. click on id=btnSend	OK	00:52:39
9. mouseOver on id=btnSend	OK	00:52:39
10. mouseOut on id=btnSend	OK	00:52:40
11. click on id=btnFeedback	OK	00:52:40
12. assertElementPresent on id=class	OK	00:52:40
'test4' completed successfully		00:52:41

This test is used to check if the feedback element is displayed after clicking the feedback button on the webpage.

Selenium IDE - TestForName*

Project: TestForName*

Executing ▾

test7* http://localhost:8080/

	Command	Target	Value
7	type	id=text	hi
8	click	id=btnSend	
9	mouse over	id=btnSend	
10	mouse out	id=btnSend	
11	click	id=btnFeedback	
12	mouse over	id=btnFeedback	
13	mouse out	id=btnFeedback	
14	click	id=feedback	
15	type	id=feedback	test
16	click	id=btnSend2	
17	assert value	id=feedback	test

Command: assert value // [icon]

Target: id=feedback [icon] [icon]

Value: test

Description:

Runs: 1 Failures: 0

Log Reference

8. click on id=btnSend	OK	02:14:22
9. mouseOver on id=btnSend	OK	02:14:22
10. mouseOut on id=btnSend	OK	02:14:23
11. click on id=btnFeedback	OK	02:14:23
12. mouseOver on id=btnFeedback	OK	02:14:23
13. mouseOut on id=btnFeedback	OK	02:14:24
14. click on id=feedback	OK	02:14:24

This test is checking the value of what the user types inside the feedback box. It checks with the asserting that both values of the user feedback are the same or not.

Video Reference

You can also view the demonisation video on this **YouTube link below**:

https://youtu.be/PPTL_FiPyEw