Submission Worksheet

IT114-006-S2024 - [IT114] Chatroom Milestone 3 2024

Submissions:

Submission Selection

1 Submission [active] 4/30/2024 12:03:53 AM ▼

Instructions

^ COLLAPSE ^

Implement the Milestone 3 features from the project's proposal
document: https://docs.google.com/document/d/1ONmvEveI97GTFPGfVwwQC96xSsobbSbk56145Xi
Make sure you add your ucid/date as code comments where code changes are done
All code changes should reach the Milestone3 branch
Create a pull request from Milestone3 to main and keep it open until you get the output PDF from
this assignment.
Gather the evidence of feature completion based on the below tasks.
Once finished, get the output PDF and copy/move it to your repository folder on your local
machine.
Run the necessary git add, commit, and push steps to move it to GitHub
Complete the pull request that was opened earlier
Upload the same output PDF to Canvas

Branch name: Milestone3

Tasks: 14 Points: 10.00

● Basic UI (2 pts.)
^COLLAPSE^

●
^COLLAPSE^
Task #1 - Points: 1

Text: Screenshots of the following

Checklist   *The checkboxes are for your own tracking
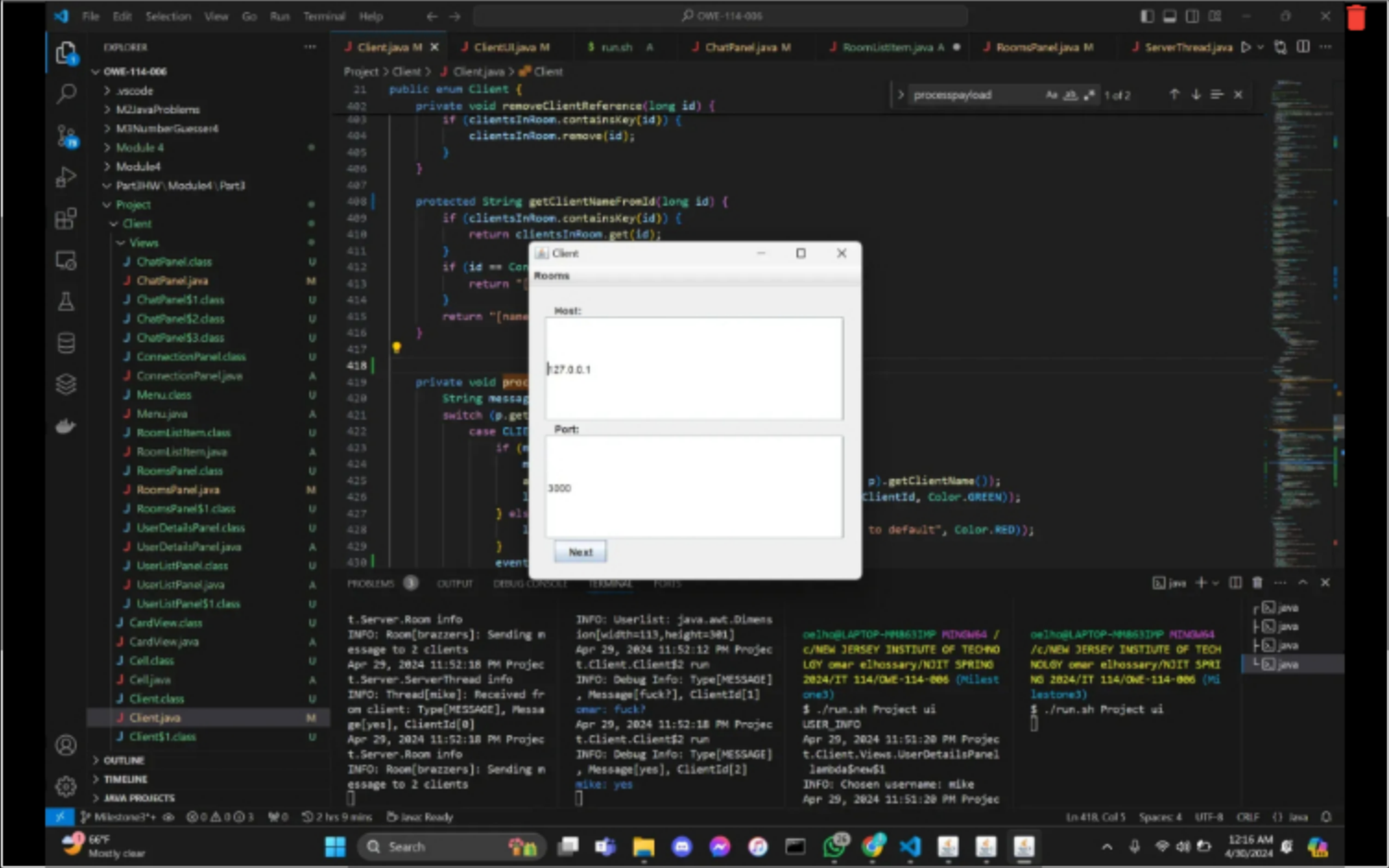
#	Points	Details

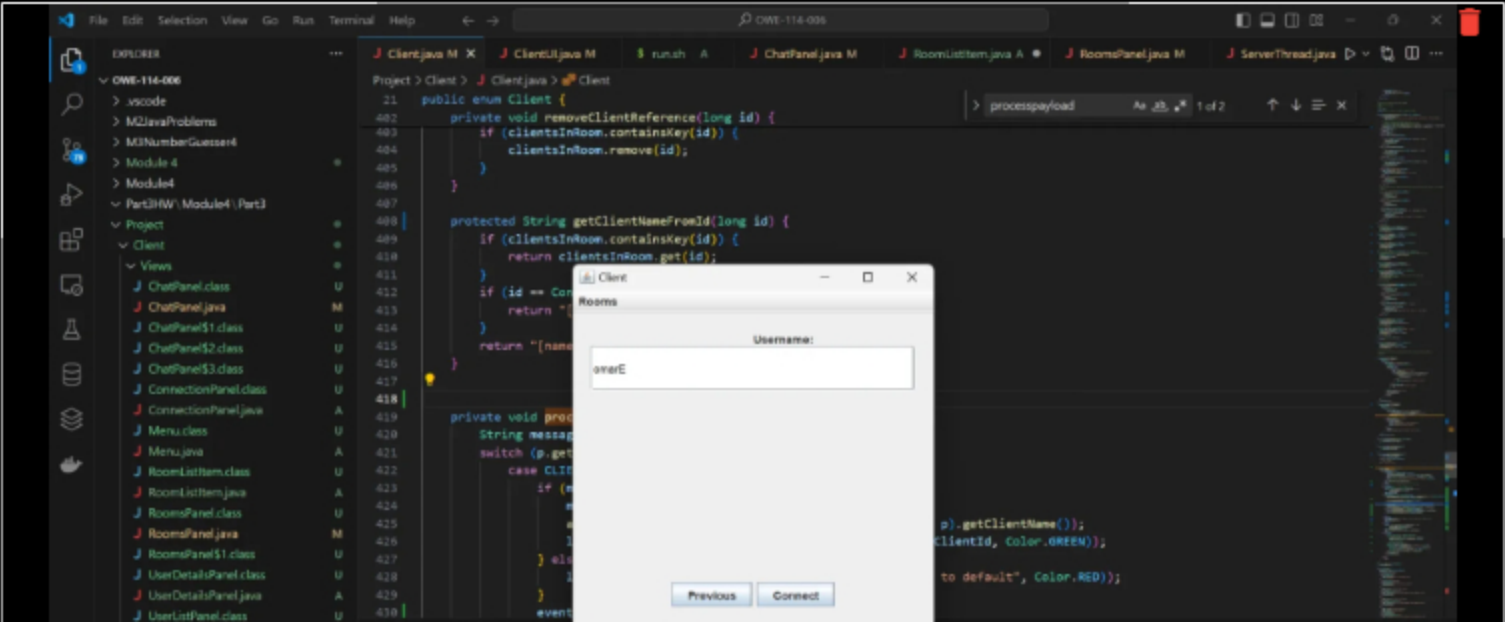|	#1	1	Connection Panel
	#2	1	User Details Panel
	#3	1	Chat Panel
	#4	1	Clearly caption screenshots

Task Screenshots:

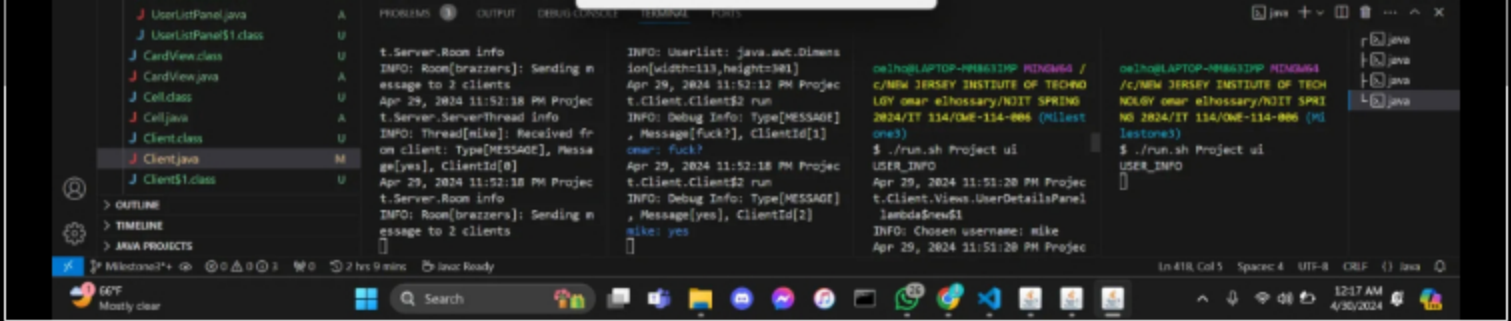Gallery Style: Large View

Small   Medium   Large
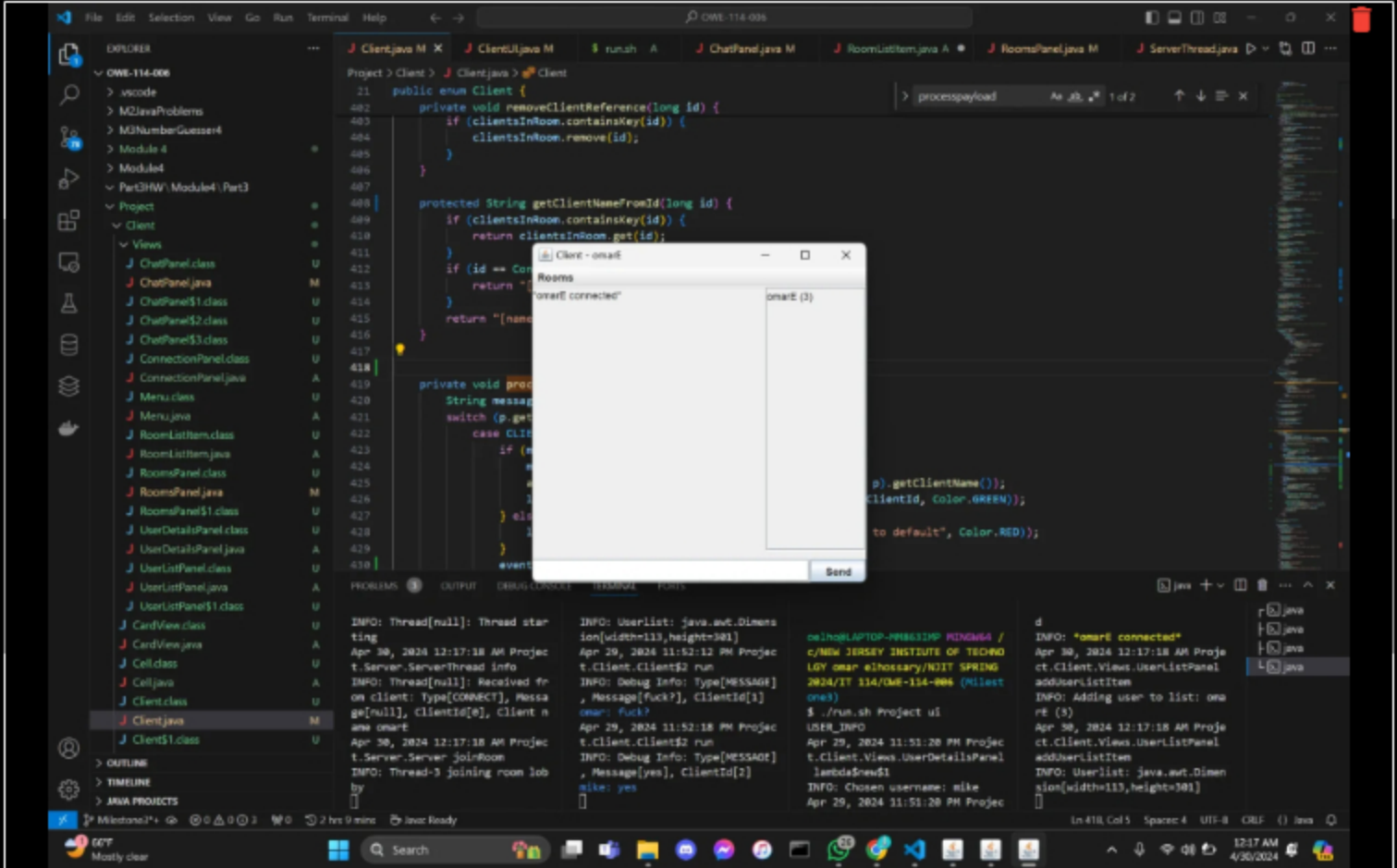

here is the connection panel

Checklist Items (0)

here is the user panel

Checklist Items (0)

and here is my successfully in the chat room

Checklist Items (0)

● Formatting (2 pts.)

^COLLAPSE ^

● Task #1 - Points: 1

^COLLAPSE ^

Text: Screenshots demoing flip and roll commands
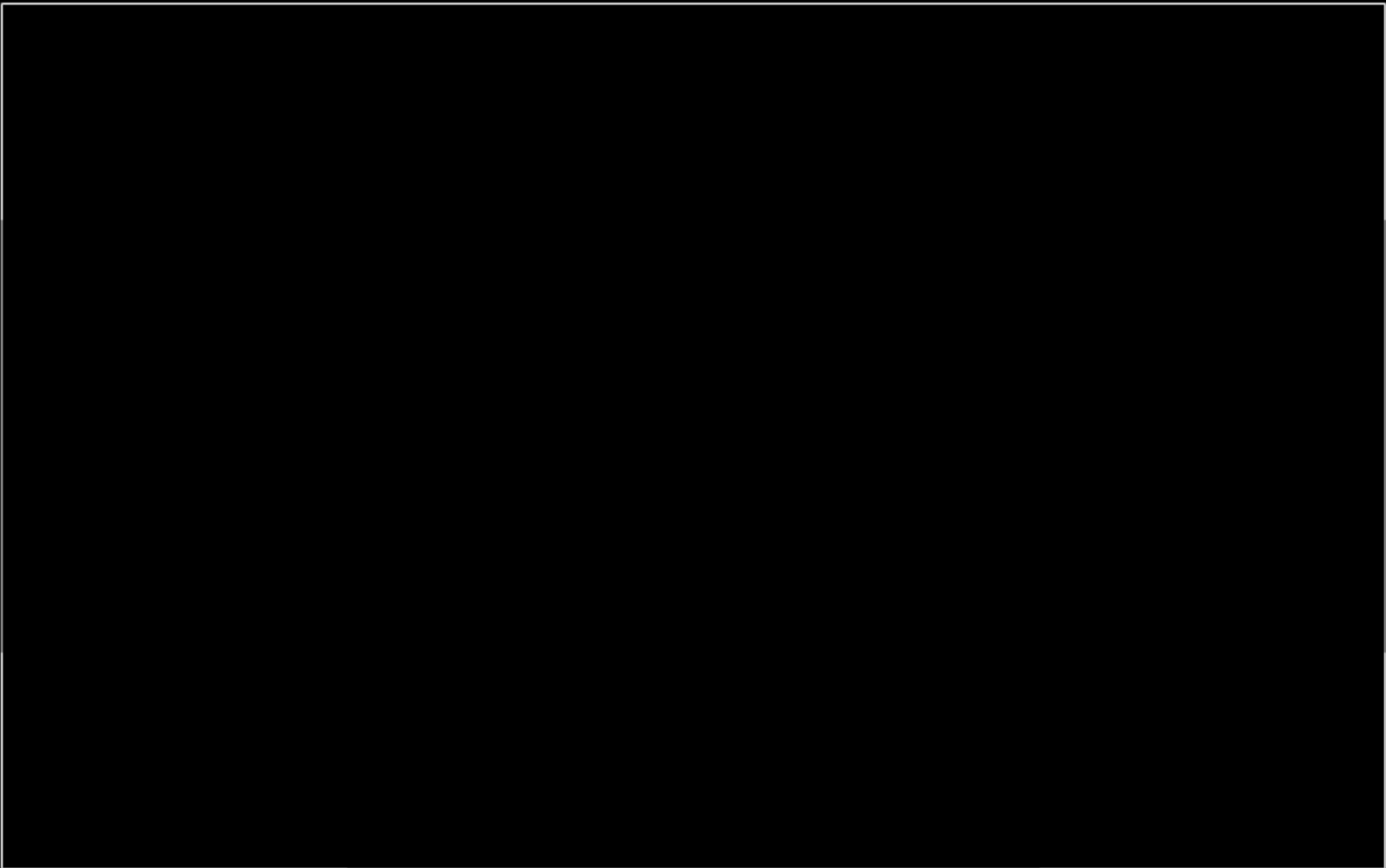
Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Flip output in a different format than normal messages

☐ #2	1	Roll # output in a different format than normal messages
☐ #3	1	Roll #d# output in a different format than normal messages
☐ #4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Missing Caption

🟢

⌃COLLAPSE ⌃

Task #2 - Points: 1

Text: Screenshots demoing custom text formatting

Checklist   *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Custom text formatting for bold working (Part of the message should appear bold)
☐ #2	1	Custom text formatting for italic working (Part of the message should appear italic)
☐ #3	1	Custom text formatting for underline working (Part of the message should appear underline)
☐ #4	1	Custom text formatting for red working (Part of the message should appear red)
☐ #5	1	Custom text formatting for blue working (Part of the message should appear blue)
☐ #6	1	Custom text formatting for green working (Part of the message should appear green)

☐ #7	1	Custom text formatting for combined bold, italic, underline, and a color working (Part of the message should have all 4 formats applied at once)
☐ #8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

here is the demonstration of the text font and color changes

Checklist Items (0)

🟢

∧COLLAPSE ∧

Task #3 - Points: 1

Text: Screenshot of the code solving the formatting display

Checklist   *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Show each relevant file this was done in (may be one or more)
☐ #2	1	Include ucid and date comment
☐ #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

here is the code for the formatting

Checklist Items (0)

Task #4 - Points: 1
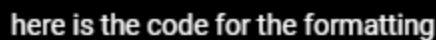
Text: Explain how the formatting was made to be visible/rendered in the UI

ⓘ **Details:**
Note each scenario

Response:

The SpecialCharacter method takes a message string and applies various formatting rules using regular expressions. It swaps out certain patterns with matching HTML tags, such as asterisks for bold, hyphens for italic, and underscores for underlining. It also manages more complicated scenarios like custom color tags and nested formatting. In the end, the technique converts unformatted text into text that is ready to be displayed in a user interface, enabling a more effective visual depiction of the message content.

● **Task #1 - Points: 1**

∧COLLAPSE∧

Text: Screenshots demoing private message

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Should have 3 clients in the same room
☐ #2	1	Demo a private message where only the sender and target see the message
☐ #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Missing Caption

Checklist Items (0)

● **Task #2 - Points: 1**

∧COLLAPSE∧

Text: Screenshots of the related code

Checklist

#	Points	Details
☐ #1	1	Show what code processes and handles the private message
☐ #2	1	The message should only be sent to the receiver and the target
☐ #3	1	The client should be targeting the username and the server side should be fetching the correct recipient
☐ #4	1	Include ucid and date comment
☐ #5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Missing Caption

Task #3 - Points: 1

Text: Explain how private message works related to the code above

^COLLAPSE^

#	Points	Details

	#	Points	Details
☐	#1	1	Include how the sender and receiver are handled
☐	#2	1	Include how the username is used to get the proper id

Response:

could not get this to work please spare me

● **Mute/Unmute Users** (3 pts.)

^COLLAPSE ^

●

^COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots demoing feature working

Checklist   *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Should have 3 clients in the same room
☐ #2	1	Demo mute preventing messages between the muter and the target
☐ #3	1	Demo mute also being accounted for with private messages
☐ #4	1	Demo unmute allowing the messages again from the target to the unmuter

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Task #2 - Points: 1

Text: Screenshots of the related code

^COLLAPSE^

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	ServerThread should have a list of who they muted
☐ #2	1	ServerThread should expose and add, remove, and is muted check to room
☐ #3	1	Room should handle the mute list when receiving the appropriate payloads
☐ #4	1	Room should check the mute list during send message and private messages
☐ #5	1	Include ucid and date comment
☐ #6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Missing Caption

Missing Caption

Task #3 - Points: 1

Text: Explain how the mute and unmute logic works in relation to the code

COLLAPSE

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Explain how your mute list is handled
☐ #2	1	Explain how it's handled/processed in send message and private message

Response:

I did not get to do this please spare me

Misc (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Add the pull request link for the branch

COLLAPSE

ⓘ Details:
Note: the link should end with /pull/#

URL #1

https://github.com/WayguBeef5/OWE-114-006/pull/10

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

COLLAPSE

Response:

I was doing this last minute and ran into multiple problems and was not able to get most things working such as the different fonts and roll and flip but will just take the points I can thank you for working with me

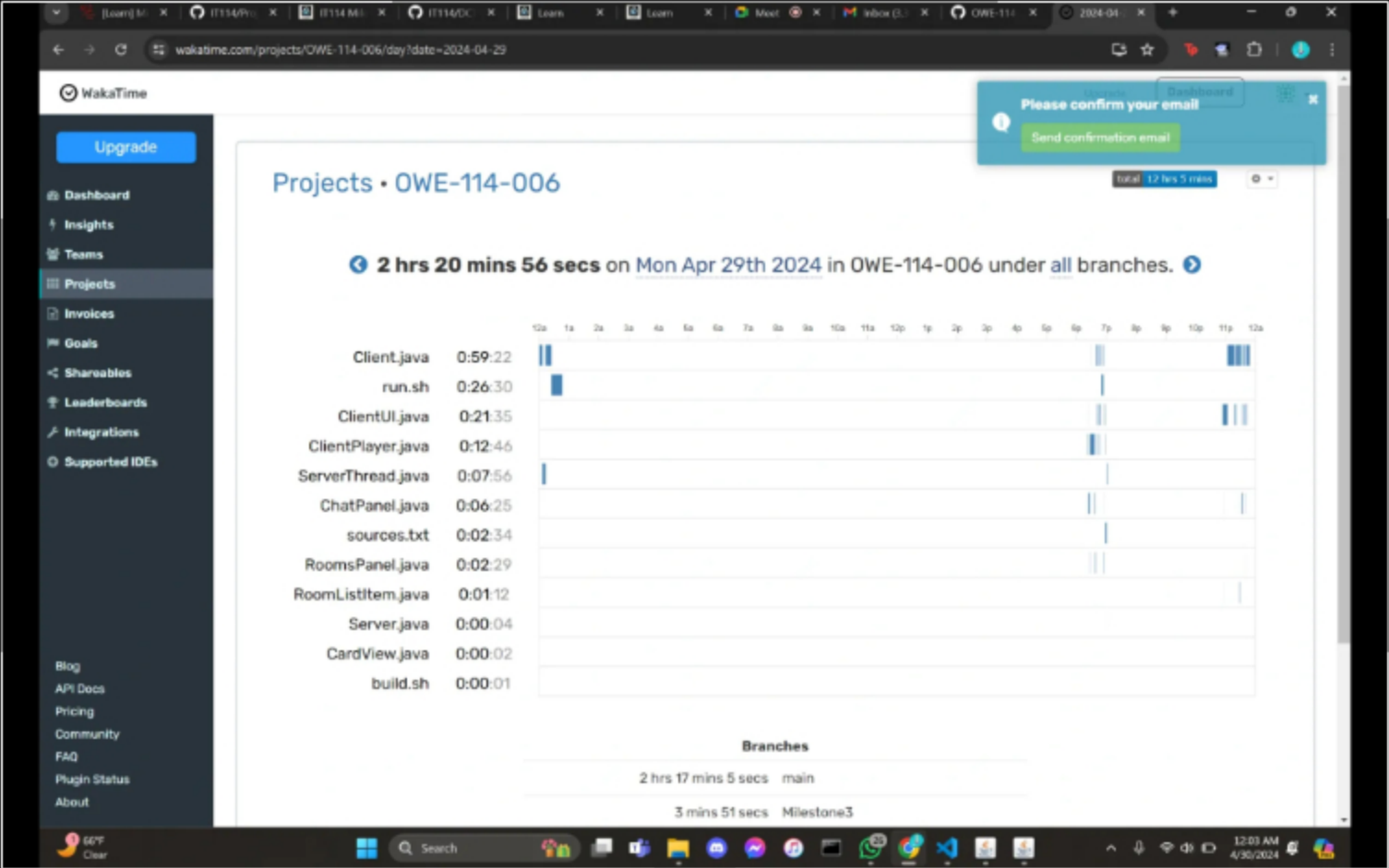## Task #3 - Points: 1

Text: WakaTime Screenshot

COLLAPSE

ⓘ Details:
Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved.

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

waka time screenshot

End of Assignment