Task 1: Exploring ChatGPT’s Role in Software Engineering

Patch Applied (PA):

For example 3 in configuring github actions workflow, chat gpt was used to better automate the building, testing, and deploying of code changes. More specifically in order to get the ci to block PRs that had labels ‘on hold’ and ‘do not merge yet’, so it cannot accidentally be merged by not reading the labels.

Notably they had a more conversion with chat gpt like after getting the code the developer asked another question to modify and explain the code chat gpt gave.

To check if the `s: on hold` label is present using the `contains` operator within a GitHub Actions workflow, you can use the `github.event.pull_request.labels.*.name` expression as follows:

yaml — Copy code

```yaml
name: Block Merge on Label

on:
  pull_request:
    types:
      - labeled
      - unlabeled

jobs:
  block_merge:
    runs-on: ubuntu-latest

    steps:
    - name: Check for blocking label
      id: check_label
      run: |
        PR_LABELS=$(jq -r '.pull_request.labels[].name' "$GITHUB_EVENT_PATH")
        if [[ $PR_LABELS == *"s: on hold"* ]]; then
          echo "This PR has the 's: on hold' label."
          exit 1
        fi
        echo "No blocking labels found"
      env:
        GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}

    - name: Unblock Merge
      if: ${{ steps.check_label.outcome == 'success' }}
      run: |
        echo "The PR does not have the 's: on hold' label. Merging is allowed."
```

In this workflow, we use `jq` to extract the labels from the pull request event payload and then check if the label "s: on hold" is present in the `PR_LABELS` variable using the `*` operator for substring matching. If the label is found, the workflow exits with a non-zero status code to block the PR from being merged. If the label is not found, the PR can be merged.

Save this YAML file in the `.github/workflows` directory of your repository, and it will take effect when a pull request is labeled or unlabeled wi[illegible]s: on hold."

But they only really used this section of the first code chat gpt gave not the 2nd one from above:
name: PR Labels
```
  - name: Check for blocking labels
      id: check_labels
      run: |
        BLOCKING_LABELS=("do NOT merge yet" "s: on hold")
        PR_LABELS=$(jq -r '.pull_request.labels[].name' "$GITHUB_EVENT_PATH")
```

```
      for LABEL in "${BLOCKING_LABELS[@]}"; do
        if [[ "$PR_LABELS" == *"$LABEL"* ]]; then
          echo "This PR has a blocking label: $LABEL"
          exit 1
        fi
      done

      echo "No blocking labels found"
```

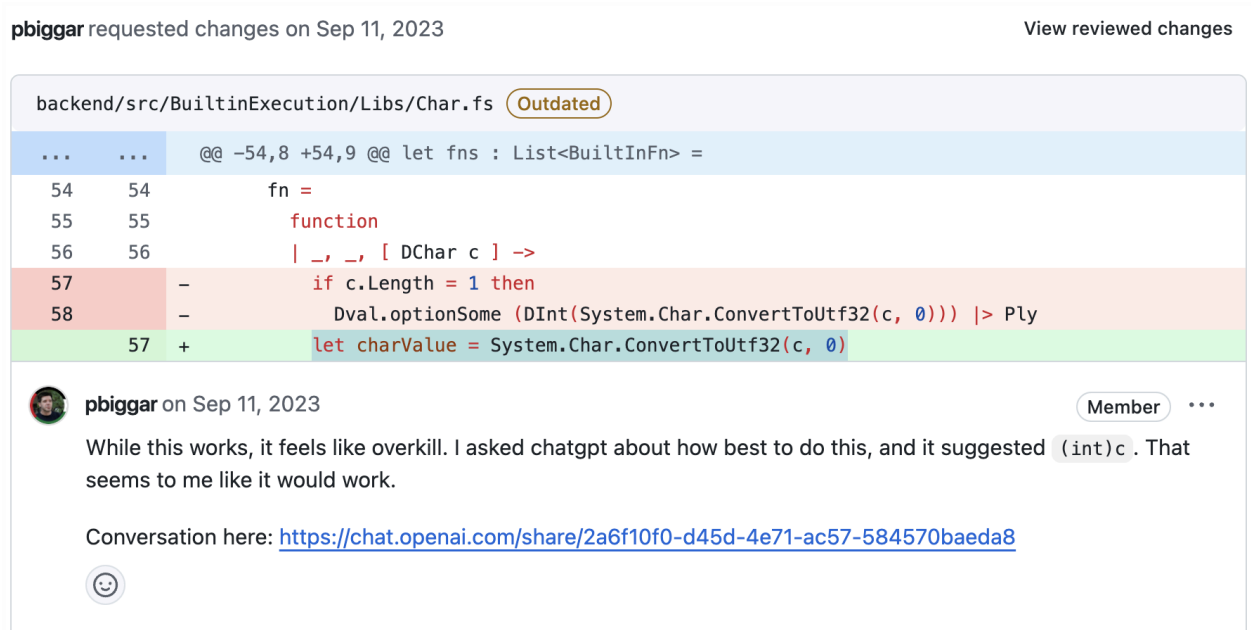
And they added more types for the pull requests like
- opened
- reopened
- synchronize
- edited

They only had to use that section of using jq to filter through json data and get labels. Chat gpt was useful in giving that idea of using jq.

Patch Not Applied (PN)

For example 2: methodological guidance, chat gpt was used to provide a more efficient way of getting the integer value of a character. The code worked before but it was used to come up with an easier approach.

Code before chat gpt was used:



Code after chat gpt was used:

```
53  56          fn =
54  57            function
55  58            | _, _, [ DChar c ] ->
56      -           (if c.Length = 1 then System.Char.IsDigit(c[0]) else false) |> DBool |> Ply
    59  +           let charValue = int c.[0]
    60  +           if charValue >= 0 && charValue < 256 then
    61  +             Dval.optionSome (DInt charValue) |> Ply
    62  +           else
    63  +             Dval.optionNone |> Ply
```

So instead of using a convert method they just cast the char to int but adding validation like checking if it is in ASCII range and other things. The code was not just taken but modified and only really that approach of casting was used. This is a clear example of methodological guidance.

None Existing Patch (NE)

For Example 1: Conceptual Guidance & Theoretical Advice the developer wanted to name variables better. Naming things is an important thing in development and is harder than you think.

Bad name for variable

app/controllers/pay.js Outdated

```
11  12      @tracked isCreatingCheckoutSession = false;
    13  +   @tracked pricingFrequencyClicked = '';
```

rohitpaulk on Dec 8, 2023 Member

Bad naming, this should be something like `selectedPricingFrequency`.

Reasoning: https://chat.openai.com/share/d668d64c-182e-4e9d-8e17-6517d91fc65e

@libmartinito we don't seem to be getting better with naming - I'd strongly suggest spending more time on this. Read books, talk to chatgpt about it, try to identify cases where our naming is wrong and ask me clarifying questions, do whatever it takes - usually these problems go away in a couple of months, doesn't seem to be happening so we need to pay extra attention to it

Better name for variable from chat gpt

```
11  13      @tracked isCreatingCheckoutSession = false;
    14  +   @tracked selectedPricingFrequency = '';
```

Chat gpt mentions “This name makes it clear that the variable holds the selected value of the pricing frequency. It's straightforward and communicates the purpose effectively.” instead of representing an action.

This is not modifying the code in its effect or even in efficiency this is just to make the code more readable and easier to understand for other people.

Task 2.1: Code Refactoring Using ChatGPT

Chat gpt conversation:https://chatgpt.com/share/6700c785-203c-800a-add4-d579b3276e4a
This is the refactored final code:
https://github.com/caesarx26/message-board-web-app/blob/main/pages/post.js

I took chat gpt's suggestions and also put a loading spinner component when the loading state is truthy. Having the validation for the post all in one function makes sense. Also adding error handling is good and using useCallback to prevent unnecessary rerenders and only when the state actually changes.

I tested the code and it works and the loading spinner shows up too.
Patch was applied.

Issue: https://github.com/David1000734/CS472TeamsRepo/issues/37
Pull request: https://github.com/David1000734/CS472TeamsRepo/pull/40

Task 2.2: Improving Documentation with ChatGPT

Chat gpt conversation: https://chatgpt.com/share/6700d763-b7e4-800a-a823-7f94e027f205
I took the documentation and changed it a bit and it works well for running and using the script.
Patch was applied

Issue: https://github.com/David1000734/CS472TeamsRepo/issues/38
Pull request: https://github.com/David1000734/CS472TeamsRepo/pull/41

Task 2.3: Understanding Complex Code with ChatGPT

Chat gpt conversation: https://chatgpt.com/share/6700d9e9-c32c-800a-9625-6bfc53998cb4
I took the explanation as is for the code and have the readme it generated as is in the pull request. Chat gpt worked well for explaining the puzzle solver.
Issue: https://github.com/David1000734/CS472TeamsRepo/issues/39
Pull request: https://github.com/David1000734/CS472TeamsRepo/pull/42

Task 2.4: Workflow Automation with GitHub Actions

I used chat gpt to improve the workflow by:

- Adding more automated tests
- Enhancing code quality checks
- Adding security analysis or vulnerability scanning
- Testing on multiple platforms or environments

Chat gpt conversation: https://chatgpt.com/share/6700e9e8-6b0c-800a-bd4e-f58da0ce6754
Issue: https://github.com/caesarx26/CS472TeamsRepo/issues/1
Pull request: https://github.com/caesarx26/CS472TeamsRepo/pull/2

Summary

Using chat gpt to help with code documentation, refactoring, and explaining code was useful. And it was helpful in reviewing other people's code as I do not know some of the frameworks or languages used for other people’s code so it gives more context in explaining the code and then you can do more research from there either by looking at documentation or asking more questions to chat gpt and the person you are reviewing the code for.