UNIVERSITÉ Concordia UNIVERSITY

SOEN-6751 HUMAN COMPUTER INTERFACE DESIGN

Instructor Dr. Rajagopalan Jayakumar

Assignment 3

Team members

Sr. No.	Name	Student ID	Email-id
1	Yash Golwala	40085663	golwalayash@gmail.com
2	Chirag Vora	40091454	chirag123192@gmail.com
3	Pooja Dhir	40104545	poojadhir1996@gmail.com
4	Harsh Divecha	40084737	harsh2826@gmail.com
5	Sanjana Udar	40094029	sanjana.udar@gmail.com
6	Jaswanth Banavathu	40080737	jaswanthbanavathu@gmail.com

Table of Contents

Table of Contents

I. Usability Testing

Usability testing in process

Usability testing is a method of testing the functionality of a website, app, or other digital product by observing real users as they attempt to complete tasks on it. The goal is to reveal areas of confusion and uncover opportunities to improve the overall user experience.

Prototype 1: StoryBoard based on User Scenario

a. User Control and Freedom

*i. **Decide What Data to Collect***

⇒ User age range determined for the user-type is accurate or needs change.
⇒ Can the user select a user-type out of its range?
⇒ How easy is it for the user to understand the error, it's type and location?
⇒ How much help is the user provided to solve the error?
⇒ Does the user like placement of various windows?
⇒ Would the user like to remove a command option from the recent window?
⇒ Would users want more command options to be added for their user-type.
⇒ Can users save and load their program in a format they would prefer?

*ii. **Run the test***

Screen 1: Selection of User Type

User 1: He found the screen pretty straightforward based on his previous experience, since the radio buttons gave him the option to select one clear choice and change the choice if it was wrong before he moved forward. Within a minute he was able to complete this task.

User 2: She shared a similar experience and had no trouble understanding the purpose of this screen. But she also pointed out that though she had 0-2 years of experience and opted for that, she had far more knowledge than a Novice user. It took her only a few seconds to complete this task.

Observers:

- It was observed that the screen was easy to use and navigate and the users didn't spend too much time here as they knew the number of years of experience they had. Thus, the response time was less.

- The user type range was not accurate as one of the users didn't fall under the category that they were directed to on the basis of their years of experience.

Screen 2: Main Screen of the GCC

User 1: The user expected to have an option to undo and redo the changes that he made to the program. He liked the fact that he could choose the commands as well as type them in or modify if needed. He would like an option to remove a command from the Recently Used Commands window.

User 2: The user appreciated the Recently Used Commands window as, though she could type or find these commands in the list of available commands, this saved her a lot of time when she wanted to reuse a command. She pointed out that it would've been helpful if she could see the errors made in the commands in real time.

Observers:

- They found new points which were missed during data collection like the option to undo and redo changes. The Recently Used Commands window was observed to be a success. They also realized the need for error handling before execution and the user might want to modify the Recently Used Commands.

*iii. **Collect, Analyze, present and comment on your data using appropriate techniques***

After Collecting the data, Observers and Designers had a meeting to analyze the flaws and the improvements required on each screen.

Screen 1: Selection of User Type

Actions by Evaluators:

1) More research and surveys need to be performed to add convincing psychological text for making the user type categories more accurate.

2) Description can be added for each user type so the user can determine more accurately the category he/she falls under.

Screen 2: Main Screen

Actions by Evaluators:

1) Add undo and redo features for editing programs on the level of words for a single step.

2) Real time monitoring of the commands being typed.

3) Provide "Remove" or "Modify" button for making changes to the Recently Used Commands list.

b. Aesthetic and Minimalist Design

*i. **Decide What Data to Collect***

⇒ Which command is used most frequently for each and amongst every user-type?
⇒ Is it easy to determine the location from where to select a command?
⇒ Can the user select multiple commands?
⇒ Can various windows be removed or added freely?
⇒ Default font style
⇒ Default color theme.
⇒ Appropriate usage of widgets that wrap a big list of options.

*ii. **Run the test***

Screen 1: Selection of User Type

User 1: The user thought that the screen was simple and had no unnecessary details. He found the screen pretty straightforward based on his previous experience, since the radio buttons acted as a good mental model for a single option selection.

User 2: She shared a similar experience and felt that the display was uncomplicated.

Observers:

- The simplicity of the screen was absolutely loved by the users and the radio buttons were a very effective mental model for a single option selection.

Screen 2: Main Screen of the GCC

User 1: He would've liked to change the theme of the smart GCC's main screen, resize the windows, and minimize or maximize the windows on this screen. He found the drop down to be convenient and made the design seem less cluttered.

User 2: She would like to see the commands in a window divided into multiple scroll-able windows according to the option type instead of the commands being available in multiple drop-down menus.

Observers:

- They realized that the Options Window had conflicting views. They also observed that the users would like a certain level for freedom to choose the look and feel of the compiler.

iii. ***Collect, Analyze, present and comment on your data using appropriate techniques***

After Collecting the data, Observers and Designers had a meeting to analyze the flaws and the improvements required on each screen.

Screen 1: Selection of User Type

Actions by Evaluators:

1) No changes are required as the design was considered to be simple and effective.

Screen 2: Main Screen

Actions by Evaluators:

1) Users can be given the freedom to change the font size, style and colour, resize the screen, minimize or maximize a window.

2) Users can be provided with a set of templates with optimized combinations of interaction metaphors.

3) The conflicting suggestions on the Options window can be incorporated in a theme.

Prototype 2: Paper Based Prototype (wireframes on paper)

a. User Control and Freedom

i. Decide What Data to Collect

⇒ Checking the ability of the user to select his/her User Type upon every visit
⇒ Interrelation between the screens
⇒ How easy it is to move from one page to another page
⇒ Changing User Type during the use
⇒ Changing commands after once selected
⇒ User able to select multiple commands
⇒ User able to enter the command not listed in his/her commands list

ii. Run the test

We ran the test for each of the screen boards in the prototype with two group members playing the role of observer and two members playing a role of participants.

Screen 1: Selection of User Type

User 1: He found design to be simple to use and had no complaint regarding selection options available with him. He took about a minute to complete this task.

User 2: She complained about not having proper guidelines for guiding her to select the appropriate category for her. She took around 3 minutes to complete the task.

Observers:

- They observed both users having no major challenge except for user2 that she was confused for a while before selecting the appropriate category for her user type and noted it to discuss that thing with designers once the test is over.

Screen 2: Main Screen of the GCC

User 1: He complained about not being able to change the user type once entered into the system. He shared his emotions for loving the drop-down list being available on the main screen which he found to be most useful. He was able to complete the task on his own on time. He was happy that he was able to enter the command which he liked and was not forced to select the command from the list.

User 2: She was happy after using the main screen and found all the suitable Interface metaphors available on screen to enhance her knowledge. Although she complained about not being able to upload the program she wanted to and had to write the program. She used multiple commands to compile and execute and loved the option of the recent commands list which saved her time for not searching and selecting the command every now and then.

Observers:

- They found interesting points when users were interacting with the designs and completed the task on time. They noted that users want to change the user type at the main screen and also users took long and made mistakes while writing the code and ask for having an option for uploading the code directly.

iii. *Collect, Analyze, present and comment on your data using appropriate techniques*

After Collecting the data, Observers and Designers had a meeting to analyze the flaws and the improvements required on each screen.

Screen 1: Selection of User Type

Actions by Evaluators:

1) More information to be added on this screen for selection of appropriate user type. This could be solved using appropriate texts for better understanding on the user side.

2) A welcome page before asking for selection of user type from the user.

Screen 2: Main Screen

Actions by Evaluators:

1) Additional button / option for changing User Type at any given point of time during the use of the application.

2) Addition of upload program or code button based on the user's feedback.

b. Aesthetic and Minimalist Design

i. Decide What Data to Collect

⇒ Checking whether the user is able to resize the window tabs displayed on the main screen of the application

⇒ Checking whether the user can read the fonts and be able to click on the button accurately

⇒ Checking whether design is easy for user to understand

⇒ Do user get an effective feedback after every action

⇒ Checking if every option available to user is visible

ii. Run the test

We ran the test for each of the screen boards in the prototype with two group members playing the role of observer and two members playing a role of participants.

Screen 1: Selection of User Type

User 1: He found the design to be simple to use as the application is giving proper feedback when the user takes the mouse over a button, the button gets highlighted. He had no complaints about the interface and was able to complete this task within a minute.

User 2: He was a bit confused with the choice as he has some experience with the application but was not sure which user type will fulfil his needs.

Observers:

- They observed both users having no major challenge except for user 2 that he was confused for a while before selecting the appropriate category for selecting the user type and he will discuss that with designers after the test.

Screen 2: Main Screen of the GCC

User 1: She was complaining about not being able to change the user type once entered into the system. A drop-down list was a good option to select the command. She was able to complete the task within a short time. She was happy that she was able to enter the command and was not forced to select the command from the list.

User 2: He liked the UI but was not happy with the option of writing the program instead of uploading a file. He liked the multiple command options and the recent command list from which he can select the commands without the need of writing it again.

Observers:

- Interesting points found when the users were interacting with the design. They found that the users wanted to have the option of uploading the file or writing the code instead of just writing it every time.

iii. *Collect, Analyze, present and comment on your data using appropriate techniques*

After Collecting the data, Observers and Designers had a meeting to analyze the flaws and the improvements required on each screen.

Screen 1: Selection of User Type

Actions by Evaluators:

1) An info page describing the categories of user before selecting user type.

2) Proper use of detailed text based on user type should be used
or the popups should be used when the user takes the cursor on a specific user type giving him proper choice to select.

Screen 2: Main Screen

Actions by Evaluators:

1) Additional button / option for changing User Type at any given point of time during the use of the application.

2) An upload feature should be there to give users a choice to upload the program.

II. Analytical Testing

Analytical Evaluation and suggestions by Inspectors

Focusing on usability, a heuristic analysis is an evaluation method in which one or more experts compare a digital product's design to a list of predefined design principles (commonly referred to as heuristics) and identify where the product is not following those principles.

"Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the 'heuristics'). — Jakob Nielsen, The Nielsen Norman Group

Prototype 1: StoryBoard based on User Scenario

a. Consistency and Standards

i. Decide What Data to Collect

⇒ Whether the Design is simple throughout the application
⇒ Whether the same design style is used for all the screens
⇒ Whether the button style and its size stay consistent
⇒ Whether the appropriate buttons and texts are used for better usability

ii. Run the Evaluation

Inspector 1 for Novice User

Screen 1: Welcome Screen

- The welcome screen has a pleasant design.

Screen 2: User Selection Screen

- The radio buttons were a convenient way of being able to select only one option while choosing a user-type.

Screen 3: Selecting Command

- The interface can be incorporated with live and frequent tutorial.
- As it could be a tedious task looking for a command, a search bar could be incorporated.
- In addition to that smart word completion feature can be done in order to complete the command rather than writing it completely and can make it easy for users.

Screen 4: Input Area

- Font size, style, colour should be in control of the user in case they are examining a large code or have a problem reading small fonts.

Screen 5: Compile

- Compile Button is active on without selecting any command which might lead to error. Also, no keyboard shortcuts found to directly compile it directly without pressing the button.

Screen 6: Output Area

- The font size, style, colour should also be adjustable here.

Screen 7: Recent Commands / All commands

- A separate window for recent commands avoids any confusion and keeps things organized on screen.

Screen 8: Exit

- At exit time, the system does not ask for the user's permission to close and all of the data is gone. So, a pop-up window should be active in order to get the confirmation from the user before exiting the GCC.

Inspector 2 for Intermediate user

Screen 1: Welcome Screen

- The buttons and typography is simple, understandable with subtle transitions like the hover effect on the button which brings the design together.

Screen 2: User Selection Screen

- Proper guidance is given using conversing text keeping no ambiguity in the user's mind and proper selection of buttons to state it subtly that the user can select only one user type at a time.

Screen 3: Selecting Command

- Users might get confused using that list which can be solved by adding a search bar for the commands.
- To keep it more engaging, addition of a smart word completion feature can be done in order to complete the command rather than writing it completely which in turn makes the program more error tolerant.

Screen 4: Input Area

- Having the control over the fonts would give the user a certain amount of freedom to read and edit the code keeping the visual comfort.

Screen 5: Compile

- A keyboard shortcut for the compile button and a pop up to confirm before the compilation would increase the ease and the error tolerance.

Screen 6: Output Area

- The font size, style, colour should also be adjustable here.

Screen 7: Recent Commands / All commands

- A separate window for recent commands avoids any confusion and keeps things organized on screen.

Screen 8: Exit

- Exit requires a pop up to confirm the termination of the program as there are changes that the user clicked on it by mistake or just remembered to make some changes. I think this is essential for error tolerance.

Inspector 3 for Expert User

Screen 1: Welcome Screen

- The text used on the welcome screen is good with a simple readable continue button which changes color over hover effect when cursor is passed over that button providing good feedback to the user.

Screen 2: User Selection Screen

- Proper guidance is given using conversing text keeping no ambiguity in the user's mind and proper selection of buttons to state it subtly that the user can select only one user type at a time.

Screen 3: Selecting Command

- Users might get confused using that list and should provide that list on the left bottom corner of the main window screen or should keep it on a separate window.
- To keep it more engaging, addition of a smart word completion feature can be done in order to complete the command rather than writing it completely and can make it easy for users and error tolerant.s

Screen 4: Input Area

- Bigger Font style option should be added as being an expert user, he might need to study a large code and the user should feel at ease while reading the code with changeable font size.

Screen 5: Compile

- Compile Button is active without selecting any command which might lead to error. Also, no keyboard shortcuts found to directly compile it directly without pressing the button.

Screen 6: Output Area

- After Compiling the code, the output area is active and there also font size should be manageable.

Screen 7: Recent Commands / All commands

- It's perfect to have recent and all commands under one tab.

Screen 8: Exit

- At exit time, the system does not ask for the user's permission to close and all of the data is gone. So, a pop-up window should be active in order to get the confirmation from the user before exiting the GCC.

iii. Collect, Analyze, present and comment on your data proposing solutions to perceived problems

After collecting the expert evaluations on prototype 1, the finalized changes required on each screen for making designs more engaging user friendly on each screen are mentioned below.

Screen	**Changes if any**
1. Welcome Screen	None
2. User Selection Screen	None
3. Selecting Command	● "Smart Word Completion Feature" with a search Bar ● Position of list to be shifted towards the left bottom side of the screen. ● Additional Guidance by providing tutorials
4. Input Area	● Bigger Font Size
5. Compile	● Shortcut for compile ● Button should be inactive if no command is selected
6. Output Area	● Bigger Font Size ● Save option must be created
7. Recent / All Commands	● Ask users to give preference for having commands in one tab or in another separate window.
8. Exit	● Warning before exiting

b. Aesthetic and Minimalist Design

i. Decide What Data to Collect

Data to be collected also represents the heuristics that will be used for evaluation of the UI.

⇒ Is simple and natural dialogue being used to guide the user?
⇒ Does the software Speak the user's language?
⇒ How many mental models have been incorporated in the design to minimize users' memory load?
⇒ How many inconsistencies in design are there across the software?
⇒ Does the software provide feedback where necessary?
⇒ Is the exit option available in all possible and necessary locations?
⇒ How many shortcuts are available in the design?
⇒ Does the UI incorporate good error messages where necessary?
⇒ Does the UI help prevent errors?

ii. Run the Evaluation

Inspector 1 for Novice User

Screen 1: Welcome Screen

- Duration for displaying the splash screen can be reduced a bit more.

Screen 2: User Selection Screen

- The dialog box is simple to understand with simple and clear text, but a hovering pop up may be required detailing some requirements to justify the range, giving a better clarity to the user.

Screen 3: Selecting Command

- Different fonts could be used for commands to differentiate from other elements.
- Fonts size could be a bit larger and color contrasting to screen could be used for ease of reading.

Screen 4: Input Area

- Good composition of space, highlighting of cursor and activation of title when clicked inside the area acts as a good feedback mechanism. Text autocomplete would be a great addition.

Screen 5: Compile

- Compile button is not so obvious, an addition of an icon or size and color change may be required.

Screen 6: Output Area

- Output area could be divided into 2 sections where one section shows all the errors encountered.

Screen 7: Recent Commands / All commands

- Latest command is visible at the bottom of the list, should be changed and made to follow a stack layout with the latest one on top.

Screen 8: Exit

- Exit button is clearly visible consistently on all pages.

Inspector 2 for Intermediate user

Screen 1: Welcome Screen

- Pleasant and serves the purpose.

Screen 2: User Selection Screen

- The dialog box is simple to understand with simple and clear text, but a hovering pop up may be required detailing some requirements to justify the range, giving a better clarity to the user.

Screen 3: Selecting Command

- The selection can be made adaptive if possible, recording and eliminating the commands that an intermediate user prefers to write rather than select more frequently.

Screen 4: Input Area

- Input area is simple and executes visibility principle clearly.

Screen 5: Compile

- Compile button is not so obvious, an addition of an icon or size and color change may be required.

Screen 6: Output Area

- The console as an output window utilizes the mental model effectively by making it obvious as a standard output area for almost all the IDEs available in the current market.

Screen 7: Recent Commands / All commands

- It is not very obvious and creates a conflict by looking almost the same as the normal command section. Some changes to the design may be necessary to make it stand out as a convenience feature.

Screen 8: Exit

- Exit button is clearly visible consistently on all pages.

Inspector 3 for Expert User

Screen 1: Welcome Screen

- Good Visibility and design standards used. Proper functioning of buttons.

Screen 2: User Selection Screen

- Good Selection of radio buttons with conversing texts.
- Proper look and feel is maintained but one alternative is to have tiles with text.

Screen 3: Selecting Command

- Design is good and proper font size and color is used.
- Can add highlights on hover effect as well before selecting any command.

Screen 4: Input Area

- Input area should be made resizable for better task completion with effectiveness.

Screen 5: Compile

- Compile button size should be changed with the logo picture of run or “!” and bigger in size.
- Color of the button can be changed to red in order to locate it easily and should be kept on the top

Screen 6: Output Area

- Output area should be made resizable and should allow the user to save the output in a file.

Screen 7: Recent Commands/ All Commands

- Proper indentation must be done.

Screen 8: Exit

- To follow standard practise to have a better feel for design, keep a close button on the top left or right corner of the screen and change is required in the color of that button.

iii. Collect, Analyze, present and comment on your data proposing solutions to perceived problems

Prototype 2: Paper Based Prototype (wireframes on paper)

a. Consistency and Standards

i. Decide What Data to Collect

⇒ The design should be consistent for all the screens.
⇒ The design should guide the user on how to use the compiler.
⇒ All options should have one type of pattern i.e. if a drop-down list is selected for compiler options then it should be there for linking options, execute options, debugging options, and all options.
⇒ All commands should work properly

ii. Run the Evaluation

Inspector 1 for Novice User

Screen 1: Selection of User Type

- The design was simple to select the user type for him. Yet, a tad of data about the compiler can be provided to him in the beginning as to why he should use this compiler rather than others.

Screen 2: Main Screen

- He was not able to run the program as his program has some errors and there was no guidance for program correction. There should be an error message where code contains an error. He was struck in one place and couldn't complete the task.

Inspector 2 for Intermediate user

Screen 1: Selection of User Type

- She was effectively ready to select the user type as she has used the compiler previously, so she didn't require any direction for that.

Screen 2: Main Screen

- There was an option to select recently used commands, but she was complaining that her previous program was not saved as there was no option to save the code.

Inspector 3 for Expert User

Screen 1: Selection of User Type

- He selected expert user types but at the same time, he was confused about how he can judge himself that now he is above the typical level.

Screen 2: Main Screen

- As already he explored all the options and everything which he used was coming in recent options, but they were not categorized, and it was taking time to find that command. He wants that in recent options all commands should come in category wise.

iii. Collect, Analyze, present and comment on your data proposing solutions to perceived problems

After Collecting the data, Observers and Designers had a meeting to analyze the flaws and the improvements required on each screen.

Screen 1: Selection of User Type

Actions by Evaluators:

1) A small pop up window with a questionnaire can be used before user selection for exactly scrutinizing the user understanding level.
2) While selecting the user type, a hyperlink can be established in the screen discussing the objectives and advantages of using the compiler.
3) Exit option should be visible to the user at all times.

Screen 2: Main Screen

Actions by Evaluators:

1) A small tutorial document can be visualized at the beginning in order to help the user of the selected type.
2) In the output area, predefined error correction techniques can be displayed in accordance to the user type.

3) A save button can be integrated to the compiler for saving the program.
4) If the application crashes, an automatic save feature must save the current state of the program in the respective user directory.

b. Aesthetic and Minimalist Design

i. Decide What Data to Collect

- Checking if the interface is showing the relevant commands for each specific type of user.
- Observing if the interface is simple without any clustering of data for the various user types.
- The interface should work fluidly without any lag when there is hierarchy and sections in the available option.
- The interface must accompany the user anytime in altering the user types with relevant information and windows.
- The interface must be clear in program execution and display of the console.

ii. Run the Evaluation

Inspector 1 for Novice User

Screen 1: Selection of User Type

- Radio buttons were used for selecting the user type giving a good design.
- One alternative is to give pop-up for each user type with details of it.

Screen 2: Main Screen

- The program window should be resizable.
- Proper fonts and messages should be displayed.
- Buttons should be highlighted once the cursor is onto it.
- File uploading option is not available.

Inspector 2 for Intermediate User

Screen 1: Selection of User Type

- Radio buttons were used for selecting the user type giving a good design.
- Exit options should be available at all times.

Screen 2: Main Screen

- The program window should be resizable.

- Proper fonts and messages should be displayed.
- Text area hover effects should be there.
- Buttons should be highlighted once the cursor is onto it.
- Output window should be resizable.
- Compiler options and other options should be available to the user while compiling the program.
- Separation of input area and output area is not available.
- File uploading option is not available.

Inspector 3 for Expert User

Screen 1: Selection of User Type

- Good Visibility and design standards used. Proper functioning of buttons.
- Good Selection of radio buttons with conversing texts
- Proper look and feel is maintained but one alternative is to have tiles with text.

Screen 2: Main Screen

- Design is good and proper font size and color is used.
- Can add highlights on hover effect as well before selecting any command.
- Input area should be made resizable for better task completion with effectiveness.
- Compile button size should be changed with the logo picture of run and bigger in size.
- Color of the button can be changed to red in order to locate it easily and should be kept on the top
- Output area should be made resizable and should allow the user to save the output in a file.
- Proper indentation must be done.
- To follow standard practise to have a better feel for design, keep a close button on the top left or right corner of the screen and change is required in the color of that button.

iii. Collect, Analyze, present and comment on your data proposing solutions to perceived problems

After Collecting the data, Observers and Designers had a meeting to analyze the flaws and the improvements required on each screen.

Screen 1: Selection of User Type

Actions by Evaluators:

1) An info page describing the categories of user before selecting user type.
2) Proper use of detailed text based on user type should be used or the popups should be used when the user takes the cursor on a specific user type giving him proper choice to select.
3) Exit option should be visible to the user at all times.

Screen 2: Main Screen

Actions by Evaluators:

1) An upload feature should be there to give users a choice to upload the program.
2) Separation of input and output area should be available.

III. Comparison between two types of testing (Usability Vs Analytical)

Differences (On Aesthetic and Minimalist Design)

Usability Testing	Analytical Testing
● Involves recording the feeling and experience of typical users' responses to different designs.	● Experts users (Inspectors) use their knowledge of users and technology to review design heuristics.
● It is done by user involvement.	● Doesn't need user involvement. Done by experts.
● Users might or might not know the criteria and would not be able to give feedback based on aesthetics and minimalist design criteria.	● Inspectors are well aware of this heuristic and can give suggestions where and when required to improvise the design.
● Cheap as it's done by the user.	● Expensive as it's done by experts.

Similarities (On Aesthetic and Minimalist Design)

Usability Testing	Analytical Testing
• User Feedback can be helpful and can lead to a design having a great UX.	• Being expert and having knowledge of user's psychology, it can also give you the suggestions which can improvise the UX of the design.
• Selection of participants from a group of users having knowledge of the aesthetics of the designs is a challenging task	• Selecting the right number of inspectors is a challenging task and also, they can be expensive.
• Provides a qualitative report after the user completes the tasks.	• Detailed report is generated after considering every aspect of the heuristics.

IV. Conclusion based on comparisons

Considering the reports and evaluations from users and inspectors; after doing thorough analysis from all of the aspects of the user control and aesthetics, Prototype 2 will be implemented for SmartGCC. Though not everything is perfect in prototype 2 design, few additional concepts and changes need to be carried from prototype 1 and from the suggestions of usability testing report and analytical testing report to make one final design which will be implemented with the best set of features.

Final Prototype: Prototype 2

Features that will be added from prototype 1:

- Conversing Texts for guidance on user type selection
- Recent / All commands as button and not as a complete list by default

List of Beneficial Features obtained from Usability Testing and Analytical Testing that can be incorporated in prototype 2 design:

- Smart Word Completion Feature
- Search bar
- More Information like Help Menu
- Font Customization

- Making compile button bigger in size and making it inactive while no command is selected
- Save option & Upload option for code
- Warning Pop Up window before exiting the application

Responsibilities & Individual Contributions

Usability Testing	Users	Testers	Observer
Prototype 1	Chirag Vora, Yash Golwala	Sanjana Udar, Harsh Divecha	Pooja Dhir, Jaswanth Banavathu
Prototype 2	Pooja Dhir, Jaswanth Banavanthu	Chirag Vora, Yash Golwala	Harsh Divecha, Sanjana Udar

Analytical Testing	Inspector	Testers
Prototype 1	Sanjana Udar, Harsh Divecha, Yash Golwala	Chirag Vora, Pooja Dhir, Jaswanth Banavanthu
Prototype 2	Chirag Vora, Pooja Dhir, Jaswanth Banavanthu	Sanjana Udar, Harsh Divecha, Yash Golwala

References

[1] R. Jayakumar, "Lecture Slides Usability Testing; Analytical Evaluation". February 2020

[2] "Usability Testing 101: What It Is & What It's NOT (2020 Guide)", *Hotjar*, 2020. [Online]. Available: https://www.hotjar.com/usability-testing/

[3] "Website Usability Testing: How To Get Started Today", *Hotjar*, 2020. [Online]. Available: https://www.hotjar.com/blog/website-usability-testing/

[4] A. Asia, "The Creative Process of User Interface Design", *The Sagara Asia Blog*, 2020. [Online]. Available: https://www.sagaratechnology.com/blog/2019/11/05/the-creative-process-of-user-interface-design/

[5] "Heuristic Analysis for UX – How to Run a Usability Evaluation", *Toptal Design Blog*, 2020. [Online]. Available: https://www.toptal.com/designers/usability/usability-analysis-how-to-run-a-heuristic-evaluation