

CS3106: Practical 1 - Heuristic Evaluation of Overleaf

Matric Number: 170002815

ABSTRACT

This report is focused on the key features and flaws the \LaTeX editor Overleaf possesses in relation to Jakob Nielsen's (1994) set of heuristics. The report discusses my individual findings from holistic evaluation, including the top positive and negative features of the editor in my opinion, as well findings from a group evaluation. A conclusion, suggesting how Overleaf could be improved, and a full table of features and flaws (Table 3) complete the report.

The motivation behind this report is to give both the reader and the writer an insight into how real-world user interfaces relate to Nielsen's heuristics, as well as providing information on how beginner evaluators should or should not approach their first evaluation.

[Abstract Word Count: 116 Words]

INTRODUCTION

Overleaf [2] is an online \LaTeX editor, designed to facilitate the creation of \LaTeX documents with the use of features such as templates, real-time previews, and in-line error messages. Online, real-time collaboration with no installation process has given Overleaf as massive user base, with over 3,900,000 students and academics, and over 3,600 institutions worldwide [2].

FIRST EVALUATION PHASE

Evaluation Environment

The evaluation was completed using Google Chrome with no plugins on a Windows 10 Home Desktop. Microsoft Edge (Windows 10) and Google Chrome (Android) were also used. These are common browser and OS choices, leading me to believe that I have completed my evaluation of Overleaf in a typical user's environment.

Evaluation Method

It was important I explored and evaluated Overleaf in a systematic way, and therefore completed my individual evaluation using the following method:

1. Study Nielsen's Heuristics, including examples.
2. Interact with Overleaf at a high level, while noting concerns and appreciations. I worked through Overleaf's interface in the following way:
 - (a) Work from the top left of to the bottom right of the page, inspecting the editor in sections.
 - (b) Complete some simple actions, such as downloading the PDF, changing small amounts of text, and making comments.

- (c) Resize & change browser. Also load on mobile device(s).
 - (d) Interact with other site pages, including the login page, register page, and home page.
 - (e) Invite a peer to experience the collaborative features of Overleaf.
3. Repeat all previous steps to catch any features or concerns I missed with the initial inspection.
 4. Review my notes created from the previous steps, assigning each concern/appreciation to at least one heuristic from Nielsen's set.
 5. Write this report in Overleaf, adding more concerns or appreciations to the previous list as they come up during my editing experience.

You can view my full table containing all features and issues I found with Overleaf's user interface in Table 3 the end of this document.

[First Evaluation Phase Word Count: 250 Words (Excluding Enumeration)]

TOP POSITIVE FEATURES

This section describes the top 5 positive features I have identified during my heuristic evaluation of Overleaf. The features are unranked, and each is accompanied by a justification for being on the list. You can view a summary of my top positive features in Table 1.1 on page 5.

File-Tree Action Buttons (Figure 1)

The buttons above the file-tree provide useful functions for the file-tree itself. On hover, the buttons show a tooltip containing the name of the action the button completes, e.g. "New File" - this helps users recognise the function of the button rather than recalling it, while also keeping the design minimalistic and attractive. The positioning of the buttons also helps users recognise their functions and associate them with the file-tree.

The button icons have excellent consistency. Internal consistency is promoted by using the same icons as those used in the file-tree itself, helping the user match the actions with the file-tree items, while external consistency is promoted by using similar icons to other programs, e.g. a folded piece of paper for creating a new file. The icons used are also similar to real-world objects, providing a match between the system and the real world.

Achieving positive remarks in four different heuristics, I believe that the file-tree action buttons are one of Overleaf's most successful user interface features in relation to Nielsen's set of heuristics.

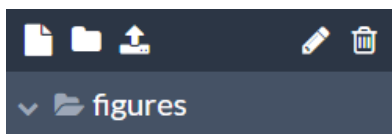


Figure 1. File-Tree Action Buttons.

Content Interaction Feedback

Overleaf provides consistent and intuitive feedback during the navigation of its various pages. The default cursor is changed to a pointer cursor for a button or action, a resize cursor for sliders, and a text cursor for text - these changes provide the user with a clear sense of what actions will be performed on a click, reducing the potential for errors and also providing external consistency with other websites.

As well as the cursor changes, the content being pointed at by the cursor also changes to provide informative feedback. Almost all items on Overleaf's site change color on hover, and most change color again on click - this creates a strong UI standard and may help the user navigate the site quicker and more confidently. Some items also provide tooltips on hover to promote recognition rather than recall, while also keeping the design minimalist.

The feedback given by content on Overleaf's pages makes exploring them an intuitive and comfortable experience, leading me to choose this feature as one of the most positive.

Error Messages

The error messages provided by L^AT_EX are unfriendly and can be quite confusing, however Overleaf has some features in relation to the user interface which facilitates the error correction process. These features have helped me decide to put error messages in my top 5 positive features list.

Colour-coding the error messages makes it easier for the user to recognise the severity of the error, and also prioritise the order in which the errors need to be dealt with. Overleaf has used red for errors and orange for warnings, which provides a match between the system and the real world.

Some error messages contain colloquial text, such as "Such boobos are generally harmless", which matches them to the real world and potentially makes it easier for users to understand the problem. Some errors even come with direct suggestions on how to fix them, such as "Just type '\l%' now", which falls into the heuristic of help users recognise, diagnose and recover from errors.

Compilation Button

The compilation button ticks many boxes in different heuristics, and I believe this is why it is one of Overleaf's top 5 features in relation to Nielsen's set of heuristics. The "Compiling..." message when the button is pressed provides the user with a good sense of system status, and the rotating icon is externally consistent with how other programs communicate a similar message.

On hover, the compilation button provides visual feedback by changing colour and showing a tooltip, promoting internal

consistency within the program. The tooltip also describes a keyboard shortcut for compilation, reminding users of the keys they need to press, rather than having to recall it themselves. The keyboard shortcut is an accelerator, giving it positive aspects in the flexibility and efficiency of use heuristic.

Help & Documentation Search (Figure 2)

The help and documentation search does not fall into as many heuristics as my other most positive features I have chosen, however it is excellent in regards to the help and documentation heuristic, leading me to put this feature on the top 5 list.

The search allows users to filter through the help and documentation that Overleaf offers without having to change pages or even use the mouse. Suggested help links are given while typing, promoting the recognition rather than recall heuristic, and many suggestions also contain a snippet of the content inside the link, making the search very efficient for use.

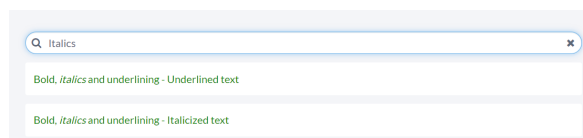


Figure 2. Help & Documentation Search.

TOP ISSUES IDENTIFIED

This section describes the top 10 issues I have identified during my heuristic evaluation of Overleaf. The issues are unranked, and each is accompanied by a justification for being on the list. You can view a summary of my top issues in Table 1.2 on page 6.

Return to Projects Button (Figure 3)

The small up arrow to the right of the menu on the editing page takes the user back to their list of projects. This arrow fails to provide external consistency or a match between the system and the real world, since up arrows tend to not mean return/back in other programs nor in every day life. The button also lacks internal consistency with the rest of the site, because it offers no feedback on hover.

I have decided to include the return to projects button in the top 10 issues list, because it has numerous negative aspects in relation to the consistency and standards heuristic and therefore fails where the rest of the site appears strong.

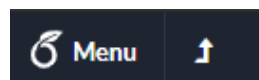


Figure 3. Menu Button & Logo (Left). Return to Projects Button (Right).

Hidden Tools

Many tools on the editor page are hidden or hard to find. A feature is unusable unless users know how to access it, and therefore I believe these hidden tools reduce user control and

freedom. These hidden tools also introduce unnecessary cognitive load for users, which hinders Overleaf in relation to the recognition rather than recall heuristic. Some examples of hidden tools include the "Find & Replace" tool, the PDF resizing tools, the edit title button, and the compilation settings.

I have included these hidden tools in my list of Overleaf's top issues, because not only do they fall short in many of Nielsen's heuristics, but even after two years of experience with Overleaf, I am still finding more tools I did not know existed.

Buttons Top Right of the Editing Page

The buttons positioned in the top right hand corner of the editing screen have many negative characteristics in relation to the set of heuristics I am using. The buttons include "Review", "Share", "Submit", "History" and "Chat" - most of these have ambiguous names and icons that are not common in other programs and do not relate to real world objects, providing external inconsistency and no match between the system and the real world. On shrinking the screen/window size, these buttons also lose their text, exaggerating the confusion.

These buttons are included in the top 10 issues list, because they could be very easily changed to have more intuitive names and icons, removing all negative aspects with very little development costs for Overleaf.

No Undo or Redo Buttons

The lack of undo and redo buttons on the editing page is extremely prominent when working with Overleaf. External consistency is broken, since other text/code editing programs have undo and redo buttons, and user control and freedom is reduced by the lack of these buttons. The common keyboard shortcuts do work, however these shortcuts are inaccessible to most mobile users, meaning errors are completely irreversible. Overleaf is one of the only text formatting programs I know to not have undo/redo buttons, leading me to put this issue on this list.

Logo Top Left of the Editing Page (Figure 3)

Logos in the top left hand corner of websites almost always provide a link back to the home page of the site, yet Overleaf does not follow this trend, prompting me to put this issue on this list. The logo on the editing page opens an options menu, making it externally inconsistent. To further this problem, the word "Menu" appears next to the logo, which some users may mistake for the button taking them to the projects menu.

No Register Form Password Check

When creating an Overleaf account, both an email and password are required - this is similar to other applications. Unfortunately, unlike other applications, Overleaf does not provide a way to check or confirm your password before creating an account - this lacks consistency and does not conform to account creation standards.

Not having a way for users to check their password decreases user control and freedom, and some users will find they need to repeat their actions to be sure they are correct. If users incorrectly enter the password they wanted, they will find

themselves locked out of their account and will need to open a different program to find the password reset link - this negatively affects Overleaf in regards to the error prevention and error recovery heuristics, and has encouraged me to put this issue on the top 10 list.

No Mobile Site

Opening Overleaf on a touchscreen device, such as an Android smartphone, loads the same website as if Overleaf is loaded on a desktop or laptop computer. This is a problem since many users will expect a mobile-friendly website, since it is a standard feature for most websites, and therefore decreases Overleaf's external consistency.

Navigating the editing page on mobile is also tricky, because only a small amount of content can be viewed at once, and the individual scrolling features for the code editing section and PDF viewing section interfere with touch navigation - this restricts user freedom and control. The lack of undo/redo buttons is also more noticeable on mobile, since few devices provide keyboard shortcuts for these features, meaning all errors are irreversible - this restricts user error recovery.

Since most issues with the site are exaggerated by loading it on a mobile device, I believe not having a mobile site is one of Overleaf's most damaging aspects to its user interface.

Content Covered by Tooltips (Figure 4)

Overleaf has excellent visual feedback when navigating the website, however some feedback is obstructive to other content. The best example of this is the tooltip which appears when hovering over the "Recompile" button, as it covers the "Auto Compile" text in the compilation settings menu. This reduces Overleaf's aesthetic design and efficiency of use, leading me to include this issue on this list.

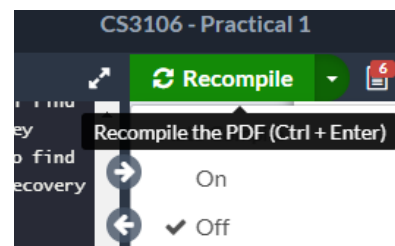


Figure 4. Tooltip Covering Content

Go to Buttons

The "Go to code location in PDF" and "Go to PDF location in code" buttons positioned between the code editor and PDF viewer are useful buttons, however they are poor UI design in relation to the visibility of system status, and the consistency and standards heuristics. When the buttons are pressed, there is no visual feedback (e.g. a colour change), which lacks internal consistency with most other buttons on Overleaf's site. The actions from the buttons also take a lot of time to be completed, yet there is no evidence to the user that the system is working - this loses the user's visibility of the system. For these reasons, I believe these buttons are one of Overleaf's biggest UI issues in relation to Nielsen's heuristics.

Ability to Undo Actions from Before System Closure

Although common keyboard shortcuts for undo/redo are available in Overleaf, there is no easy way for users to undo their actions they completed before the system is closed (e.g. reloading website). An example of this is when a user may accidentally delete a section of their code and then lose connection; when the site reloads, their accidental deletion cannot be reversed. This reduces Overleaf's user control and freedom, as

well as its ability to help users recover from errors.

Since a lot of work completed by users can be lost because of this issue, I believe it deserves to be included as one of the top 10.

Table 1.1: Top Features Summary Table

Identified Feature	Related Heuristic	Justification
File-Tree Buttons	Recognition Rather than Recall	Provide good tooltips on hover
	Consistency and Standards	Icons which promote external consistency (e.g. folded paper for new file) & also internal consistency (e.g. new file is same icon as file).
	Match Between System and Real World	Icons are relatively intuitive and relate to real-world objects.
Content Interaction Feedback	Consistency and Standards	All buttons except "Rich Text" and "Source" have a response on hover (change color), and several also show a tooltip. Most buttons have a response on click (change color again).
Error Messages	Recognition Rather than Recall	Error messages cause notification number to appear next to link to find them.
	Help Users Recognize, Diagnose and Recover from Errors	Error messages are easy to locate, informative and are colour-coded to make things easy for the user to debug.
	Match Between System and Real World	Error messages contain colloquial text to explain the actions the user needs to take, rather than just LaTeX error messages. E.g. "Such boobos are generally harmless so keep going".
Compilation Button	Visibility of System Status	Message to explain system's current action & also a stylish progress bar.
	Consistency and Standards	Good: Ctrl + S (on Windows/Linux) causes the system to compile the code, rather than saving the webpage. This holds up external consistency with other text editing programs, where users may use Ctrl + S to save the document.
	Flexibility and Efficiency of Use	Easy keyboard shortcut to be used as an accelerator, but the button is still easy to access for both experienced and beginner users.
	Recognition Rather than Recall	On hover of the "Recompile" button, a tooltip suggests the keyboard shortcut to be used as an accelerator. This promotes recall for intermittent users who may forget the shortcut.
Help & Documentation Search	Help and Documentation	Provides an easy way to traverse the documentation. Many search results are questions, which are easy for users to see if their question will be answered on that page.
	Recognition Rather than Recall	Suggestions provided while typing, which user can recognise.

Table 1.2: Top Issues Summary Table

Identified Issue	Related Heuristic	Justification
Return to Projects Button	Match Between System and the Real World	Up arrows are rarely ever associated with return to home or previous location in real world, or in other technology situations.
	Consistency and Standards	Up arrows are rarely ever associated with return to home in other UIs. No tooltip like other buttons on this site.
Hidden Tools	Recognition Rather than Recall	Many tools are hidden or hard to find in the editor and are unusable unless users can recall how to access them.
	User Control and Freedom	Same reasoning as above.
Buttons Top Right of Editing Page	Match Between System and the Real World	Icons and names have many different interpretations, which could be confusing to the user. Buttons also lose text when screen size is reduced, leading to even more confusion.
	Consistency and Standards	Most names ambiguous and are not common in other programs.
No Undo or Redo Buttons	User Control and Freedom	Keyboard shortcuts for undo/redox work as expected, however there are no clickable options to complete these actions. This problem is exaggerated on mobile devices which do not commonly have undo or redo shortcuts.
	Consistency and Standards	Most other programs have undo/redox buttons.
Logo Top Left of the Editing	Consistency and Standards	Most logos return the user to the home screen, however this logo opens the menu.
No Register Form Password Check	Error Prevention	Register form has no way to check password (e.g. input match or reveal button), so input errors are unknown and difficult to reverse.
	Consistency and Standards	Most other programs have password checks.
No Mobile Site	Flexibility and Efficiency of Use	Loading the LaTeX editor on mobile loads a desktop site. This site is difficult to traverse and use, especially for a small mobile device.
	Consistency and Standards	Most other websites have mobile sites.
Content Covered By Tooltips	Aesthetic and Minimalist Design	Some tooltips cover other important text.
Go To Buttons	Visibility of System Status	Buttons work correctly, however the actions are not immediate and there is no response to show that the system is working.
Ability to Undo Actions from Before System Closure	Help Users Recognize, Diagnose, and Recover from Errors	Mistakes made shortly before system close are difficult to undo, especially if they are on a large scale and no version has been downloaded.
	User Control and Freedom	Same reasoning as above.

SECOND EVALUATION PHASE

The second phase of my heuristic evaluation of Overleaf involved comparing my results from the first phase with other students. To begin, a time and location for the group to meet was agreed upon using Facebook Messenger [1], and everyone arranged to have their individual evaluation completed before this time.

Once the group had gathered, we began comparing results using the following method:

1. Loop through Nielsen's set of heuristics and complete the following actions:
 - (a) Each group member reads out their positive features and negative issues from their full table.
 - (b) After each feature/issue is read out, the other group members raise their hands if they have a similar feature/issue in their table.
 - (c) Both the reader and those who raised their hands note down how many agreements there were for that feature/issue.
 - (d) If a large number of group members have something similar listed in their full table, or if the entire group agrees it was a good point, the feature/issue is noted down.
2. Once all features and issues are read out and agreements are noted, the top 10 issues and top 5 features from those noted down in step (1d) are chosen.
3. These top 10 issues and top 5 features are then discussed to make sure the group agrees on the exact problems or benefits to be written about in the report.

You can view my aggregated summary tables containing the top 5 features and top 10 issues my group agreed upon in Table 2.1 (Page 9) and Table 2.2 (Pages 10 & 11) respectively.

[Second Evaluation Phase Word Count: 217 Words (Excluding Enumeration)]

AGGREGATED TOP POSITIVE FEATURES

This section describes the top 5 positive features agreed upon with my group during the second evaluation phase. The features are unranked and each is accompanied by a justification for being on the list. You can view an aggregate summary of my group's top positive features in Table 2.1 on page 9.

Iconography

Every group member had a positive comment about Overleaf's iconography, and many had a related feature in their top 5 positive features list, leading us to include it as one of the agreed upon top features. As described in my *File-Tree Action Buttons* and *Compilation Button* subsections, many of the icons and images used on Overleaf's site are very intuitive and promote both internal and external consistency, as well as a match between the system and the real world. Many icons offer a way for users to recognise the actions associated with different buttons rather than having to recall them, and some help display the systems status in a clear manner to the user, all while keeping the design of the site minimalistic and aesthetically pleasing.

Syntax Suggestions

L^AT_EX has a huge library of commands, and therefore by offering suggestions of snippets in real-time, Overleaf makes the L^AT_EX coding experience much more user-friendly. Providing syntax suggestions gives an excellent way for users to recognise commands rather than having to recall them, and also improves Overleaf's efficiency of use by letting users select a command or code snippet without having to type it out. Not all group members had something in their full or summary table about this feature, however it was agreed that is one of Overleaf's best features in regards to its user interface.

Keyboard Shortcuts

Overleaf provides a multitude of keyboard shortcuts, including "Ctrl + Enter" for compilation, "Ctrl + Z" for undo, and "Ctrl + F" to open the find and replace tool (Windows 10). These keyboard shortcuts act as accelerators and provide users with more control and freedom, and also provide a more flexible and efficient use of Overleaf. Most group members mentioned something positive about keyboard shortcuts in their summary tables, and all members agreed they used the shortcuts on regular occasions while using Overleaf - this explains its place on the aggregate top features list.

Minimalistic Design

Although not every aspect or feature of Overleaf is minimalistic and aesthetically pleasing, overall the website is attractive and has a focused design. During the group discussion, the high signal-to-noise ratio of the editing page was brought up on several occasions, and all members enjoyed the internally consistent colour scheme associated with all of Overleaf's pages, placing this feature in the top 5 list.

Error to Code Location

Few group members had noticed that clicking on an error or warning brings the user to the exact line of code causing it, however all members agreed that this user interface feature fits positively into many of Nielsen's heuristics, encouraging us to put it in the top 5 features list. Removing the need to remember line numbers decreases users' cognitive load, which fits into the recognition rather than recall heuristic; removing the need to traverse the code to find the error improves the flexibility and efficiency of use, and also helps users diagnose and recover from errors.

AGGREGATED TOP ISSUES

This section describes the top 10 issues agreed upon with my group during the second evaluation phase. The issues are unranked and each is accompanied by a justification for being on the list. You can view an aggregate summary of my group's top issues in Table 2.2 on pages 10 and 11.

Poor Timing of Error Production

Overleaf fails to point out most syntax errors before users compile documents, decreasing Overleaf's error prevention. Only simple errors such as missing brackets are corrected in real-time, and this decreases external consistency with other

code editors, most of which find many mistakes as soon as they are input into the code. All group members agreed they would enjoy an improved error recognition system, and so this issue was added to the top 10 list.

No Undo or Redo Buttons

Every group member noticed the lack of undo and redo buttons on the editing page, and this point was also contained in every summary table, leading us to include it in the top 10 issues list. The reasons for this issue being a major problem in relation to Nielsen's heuristics is described in the *Top Issues Identified* section of my report.

Hidden Tools

Most group members noted at least one tool being hidden from users on the editing page, encouraging this issue to be included in the top 10 list. The most common tool recognised was the edit title button, which is only revealed when hovering the title. The reasons for this issue being a major problem in relation to Nielsen's heuristics is described in the *Top Issues Identified* section of my report.

Fewer Keyboard Shortcuts in Rich Text Mode

Only one out of four group members recognised that fewer keyboard shortcuts worked in Rich Text editing mode, but all group members agreed this had negative effects in many of Nielsen's heuristics. The lack of transferable skills from the Source to Rich Text editing modes reduces Overleaf's internal consistency, and flexibility and efficiency of use. Users cognitive load is increased by the lack of keyboard shortcuts, as they need to remember how to complete actions without using accelerators they may be used to. Because this issue causes Overleaf to fall short in many of Nielsen's heuristics, it has been included in the top 10 list.

Menu

A considerable number of errors were brought up in relation to the menu in Overleaf's editor. Most group members noticed an issue with the logo and menu button having a confusing action, as described in the *Top Issues Identified* section of this report, however many members also noticed issues with the menu itself.

The menu lacks consistency with itself, as different sections are displayed in different ways, and the "Hotkeys" section brings up an entirely new menu for itself. The menu is very small, even though the rest of the page cannot be accessed while it is open, and there is no close button to be found - both of these factors decrease Overleaf's user control and freedom, as well as external consistency with other programs and websites. The external consistency of the menu is reduced even further, because the common actions found in menus (e.g. "Save File" or "Open File") are nowhere to be seen.

The multitude of problems relating to Overleaf's menu lead the group to include it as one of the top 10 issues.

Errors

The timing of errors has already been mentioned in this list, however the group agreed that the errors themselves are a huge

UI problem for Overleaf. Most errors are hidden away behind a tiny icon with no text, while other errors are very aggressive and cover the majority of the screen with a red highlight - both of these situations can be confusing and scary to users, which negatively affects Overleaf's aesthetic and minimalist design, user control and freedom, and error prevention system.

Alongside the counter-intuitive visual cues for errors, Overleaf provides poor error content, with unusual scales such as "Badness" and unhelpful messages like "If spelt incorrectly, spell it correctly", which reduce the match between the system and the real world, and error recovery respectively.

Similar to the menu, the sheer number of problems relating to Overleaf's errors lead the group to include them as one of the top 10 issues.

No Register Form Password Check

Only half of the group members noticed this issue when completing their personal evaluations, however all group members agreed this was a terrible design flaw on Overleaf's behalf, leading to this issue being included in the top 10 list. The reasons for this issue being a major problem in relation to Nielsen's heuristics is described in the *Top Issues Identified* section of my report.

Poor Automatic Error Handling

In combination with error content and error timing, Overleaf tries to correct some errors automatically without notifying the user, usually with negative effects on the users' work. Correcting only some errors automatically is inconsistent and users have no way to tell when Overleaf has done this, which reduces user control and freedom. Alongside holding Overleaf back in relation to many heuristics, all group members found this issue annoying, and therefore it is included in the top 10 list.

Return to Projects Button

During the group discussion, this issue came up the most frequently and also appeared in the most heuristics out of all of the errors discussed, giving us no choice but to include it in the top 10 list. The reasons for this issue being a major problem in relation to Nielsen's heuristics is described in the *Top Issues Identified* section of my report.

No Link to Documentation

A huge UI flaw which I did not notice during my personal evaluation is the lack of a link to the Help & Documentation pages from the editing page. Most programs offer an easy way to get to the necessary documentation, and Overleaf not including this feature reduces its external consistency.

Users will find it more difficult to access the help/documentation they need, which in turn worsens Overleaf's UI in relation to the help and documentation, and help users diagnose and recover from errors heuristics. To make things worse, Overleaf's project page includes a "Help" button, reducing the internal consistency between the different pages.

Table 2.1: Top Aggregate Features Summary Table

Identified Feature	Related Heuristic	Justification
Iconography	Consistency and Standards	Icons used promote external consistency (e.g. folded paper for new file) & also internal consistency (e.g. new file is same icon as file).
	Recognition Rather than Recall	Icons paired with recognisable actions.
	Match Between System and Real World	Icons are relatively intuitive and relate to real-world objects.
	Visibility of System Status	Some icons imply system status (e.g. spinning compile icon)
Syntax Suggestions	Recognition Rather than Recall	Suggested snippets allow users to skip typing most commands.
	Flexibility and Efficiency of Use	Same reasoning as above.
Keyboard Shortcuts	User Control and Freedom	Shortcuts act as accelerators.
	Flexibility and Efficiency of Use	Users have a choice for how actions are completed - keyboard or mouse.
Minimalistic Design	Aesthetic and Minimalist Design	Focused design, with a high signal-to-noise ratio.
	Consistency and Standards	Site-wide consistency with regards to the colour scheme.
Error to Code Location	Flexibility and Efficiency of Use	Provides an easy way to traverse the documentation. Many search results are questions, which are easy for users to see if their question will be answered on that page.
	Recognition Rather than Recall	Suggestions provided while typing, which user can recognise.

Table 2.2: Top Aggregate Issues Summary Table

Identified Issue	Related Heuristic	Justification
Poor Timing of Error Production	Error Prevention	Most errors not found until compilation.
	Consistency and Standards	Most other code editors that provide error correction catch more errors than only bracket matching.
No Undo or Redo Buttons	User Control and Freedom	Keyboard shortcuts for undo/redo work as expected, however there are no clickable options to complete these actions. This problem is exaggerated on mobile devices which do not commonly have undo or redo shortcuts.
	Consistency and Standards	Most other programs have undo/redo buttons.
Hidden Tools	Recognition Rather than Recall	Many tools are hidden or hard to find in the editor and are unusable unless users can recall how to access them.
	User Control and Freedom	Same reasoning as above.
Fewer Keyboard Shortcuts in Rich Text Mode	Flexibility and Efficiency of Use	No keyboard shortcuts available in Rich Text mode means fewer ways to complete different actions.
	Consistency and Standards	Source mode has different shortcuts, creating a lack in internal consistency.
	Recognition Rather than Recall	Users need to remember how to use buttons instead of shortcuts they may have learned.
Menu	Consistency and Standards	Different sections displayed differently, breaking internal consistency. Lacks similarity to other websites (e.g. close menu button)
	User Control and Freedom	No close button.
Errors	Error Prevention	Errors are intimidating and hard to understand/find.
	Aesthetic and Minimalist Design	Some errors are far more ugly and aggressive than necessary.
	User Control and Freedom	Errors hidden away behind a tiny icon.
	Help Users Recognize, Diagnose, and Recover from Errors	Some error messages difficult to understand, and some give wrong/unhelpful advice.
	Match Between System and Real World	"Badness" scale not related to real world.

Table 2.2 (Continued)

No Register Form Password Check	Error Prevention	Register form has no way to check password (e.g. input match or reveal button), so input errors are unknown and difficult to reverse.
	Consistency and Standards	Most other programs have password checks.
Poor Automatic Error Handling	Help Users Recognize, Diagnose, and Recover from Errors	Fails to fix errors, but still hides them behind a tiny icon.
	Consistency and Standards	Only some errors are "corrected", and user has no way to find out which ones.
	User Control and Freedom	Same reasoning as above.
Return to Projects Button	Match Between System and the Real World	Up arrows are rarely ever associated with return to home or previous location in real world, or in other technology situations.
	Consistency and Standards	Up arrows are rarely ever associated with return to home in other UIs. No tooltip like other buttons on this site.
No Link to Documentation	Help Users Recognize, Diagnose, and Recover from Errors	No way for users to get to help directly from making an error.
	Consistency and Standards	Other sites have help buttons readily available (no external consistency) & Overleaf's project page has a help button (no internal consistency)
	Help and Documentation	Difficult to access help and documentation.

CONCLUSION

Overall, Overleaf is an excellent \LaTeX editing program with an attractive user interface that has many positive features. Nielsen's set of heuristics reflect this well, with over 70% of all the features/issues I noticed being positive. However, in my opinion, there are some small changes Overleaf could make to improve the UI and have it reflect more positively in relation to Nielsen's heuristics.

Many issues identified during the first and second stage of evaluation mentioned the menu button and the return arrow positioned at the top left of the editing screen. Overleaf could remove the up arrow, and separate the menu button from the logo; the menu button could then be renamed to "Toolbar" and open a much larger overlay than is currently provided, while the logo can now link to the home/projects page. These changes would improve Overleaf's UI in relation to the user control and freedom, consistency and standards, recognition rather than recall, and aesthetic and minimalist design heuristics, all without requiring much development.

Two very simple changes that Overleaf could make to their website would be to include a "Help" button on the editing page and a password check on the register form. The help button could be copied from the projects page and would improve all heuristics relating to errors, while the password check would conform with account creation standards and help prevent errors.

Overleaf's errors pose a large problem for its UI. Some small changes to reduce their negative impact could be to lessen the amount of code highlighted for missing brackets, and also provide a more obvious button for users to open the error section of the editor with. The content of Overleaf's errors could also be improved, however this would most likely require far more development time than my other suggestions.

[Conclusion word count: 300 Words]

PERSONAL EVALUATION & REFLECTION

Being my first heuristic evaluation, this project has been an interesting learning experience for me. I believe I have been successful in understanding each of Nielsen's ten heuristics, and have also been successful in achieving everything required by the specification, including the organisation and coordination for the group oriented section of this project.

In terms of downfalls, I noticed other group members had picked up on some seemingly obvious issues with Overleaf's UI which I missed, and I also found it difficult to draw a line between positives/negatives with \LaTeX and positives/negative with Overleaf.

If I were to complete this project again, I would spend more time with my initial evaluation, and work through each of my given heuristics in order, questioning each feature of Overleaf, rather than finding features then comparing them to different heuristics. By working this way round, I believe I would find a more complete set of features/issues which relate more closely to my given heuristics.

REFERENCES

- [1] Messenger. 2019. Facebook Messenger. (2019). Retrieved October 14, 2019 from <https://www.messenger.com>.
- [2] Overleaf. 2019. Overleaf, Online LaTeX Editor. (2019). Retrieved October 9, 2019 from <https://www.overleaf.com>.

Table 3: Full Features/Issues Table

List of Heuristics	Identified Feature or Problem	Brief Justification	Agreement (Max: 3)
Visibility of System Status	Loading Splash Screen	Good: Message to explain system's current action & also a stylish progress bar.	0
	Recompile Loading Icon	Good: Recompile button has immediate response and also shows user that compilation is happening with text and a repeating icon.	2
	Go To Location Buttons	Bad: "Go to code location in PDF" and "Go to PDF location in code" buttons work correctly, however the actions are not immediate and there is no response to show that the system is working.	1
Match Between System and the Real World	Real Time Collaboration	Good: Collaborating users have individual cursor and changes can be seen in real time. This is similar to watching someone else edit work in a real world situation.	1
	Login Method Icons	Good: Familiar icons are used for different login methods (e.g. Twitter logo used for login with Twitter), promoting external consistency, and the match between the system and the real world.	3
	File-Tree Icons	Good: All icons in the file tree are relatively intuitive and relate to real-world objects.	3
	Error Messages	Good: Error messages contain colloquial text to explain the actions the user needs to take, rather than just LaTeX error messages. E.g. "Such booboos are generally harmless so keep going".	1
	Comments	Good: Comments relate to a highlighted section, which is similar to a physical comment/highlight in a book or essay.	0
	Rich Text Mode	Good: Allows LaTeX code to be viewed in a version closer to the final product, and therefore closer to the real world.	0
	Buttons Top Right of Screen	Bad: Icons and names have many different interpretations, which could be confusing to the user. Buttons also lose text when screen size is reduced, leading to even more confusion.	1
	Up Arrow to Go Home or to Projects	Bad: Up arrows are rarely ever associated with return to home or previous location in real world, or in other technology situations.	1
User Control and Freedom	Buttons Top Right of Screen	Good: If pressed without intention of completing action, pressing the button again takes user back to main editing screen.	0
	Simple Text Editing	Good: Code editor is very intuitive and has almost zero restrictions on what and where you can type.	0
	Section Resize Sliders	Good: User has full control over size and proportions of the file-tree, code editor, chat, and PDF viewer.	0
	Code & PDF Traversal	Good: Code editor and PDF viewer have individual scroll bars which work on hover, and the editor offers many ways to get to a section of code or section of the PDF ("Go To" buttons, double clicking & a dedicated find and replace tool).	1
	Hidden Tools	Bad: Many tools are hidden or hard to find in the editor and are unusable unless users know how to access them. Examples include Find & Replace tool, PDF resizing, hiding the file-tree, edit title button, and compilation settings. This also crosses over with the "Recognition over Recall" heuristic.	2
	Comments	Bad: Can only make comments when code is selected. This is minimalistic, but unintuitive.	0
	Errors	Bad: A few rare errors had no easy way out and caused me to refresh the page.	0
	No Undo/Redo Buttons	Bad: Keyboard shortcuts for undo/redo work as expected, however there are no clickable options to complete these actions. This problem is exaggerated on mobile devices which do not commonly have undo or redo shortcuts.	2
	Register Emergency Exit	Bad: Register form has no emergency exit, trapping the user on this page without the use of the browser buttons.	0
	Action History Cleared on Refresh	Bad: Mistakes made shortly before system close are difficult to undo, especially if they are on a large scale and no version has been downloaded.	0

Table 3 (Continued)

Consistency and Standards	Login Method Icons	Good: Familiar icons are used for different login methods (e.g. Twitter logo used for login with Twitter), promoting external consistency, and the match between the system and the real world.	0
	Action Buttons	Good: Site wide internal consistency of green buttons with white text used for almost all action buttons.	1
	Double Click Title to Edit	Good: Since matches with most other text editing software/UIs.	0
	File-Tree Action Buttons	Good: Use icons which promote external consistency (e.g. folded paper for new file) & also internal consistency (e.g. new file is same icon as file).	0
	Cursor Changes	Good: Pointer cursor for action buttons/objects, text cursor for selectable text, resize cursor for resizing sections, and default cursor for everything else upholds internal and external consistency.	0
	Line Numbers	Good: Code editor has line numbers, which is consistent with other code editors (externally consistent).	0
	Button Interaction Responses	Good: All buttons except "Rich Text" and "Source" have a response on hover (change color), and several also show a tooltip. Most buttons have a response on click (change color again).	0
	Error Notifications	Good: Error notifications uses a small number in the top right corner of the icon to link to the error section. This promotes external consistency, for example Android's app notifications.	0
	Common Save Shortcut = Compile	Good: Ctrl + S (on Windows/Linux) causes the system to compile the code, rather than saving the webpage. This holds up external consistency with other text editing programs, where users may use Ctrl + S to save the document. This also helps prevent errors.	1
	Resizing Bars	Good: Work similar to other computer programs, e.g. Photoshop. This promotes external consistency.	0
	Inter-Section Consistency	Good: Both PDF viewer and code editor feel similar and have similar actions - this promotes internal consistency.	1
	No Undo or Redo Buttons	Most other programs have undo/redo buttons.	2
	Down Arrow Beside File-Tree Selection	Bad: Loses internal consistency since arrow is same icon as the icon used for expanding a folder in the file-tree. Loses external consistency since down arrow rarely ever represents options in websites.	0
	Buttons Top Right of Editing Page	Most names ambiguous and are not common in other programs.	0
	Logo on Editor	Bad: Most logos return the user to the home screen, however this logo opens the menu. Some recovery in that the menu is highlighted on logo hover.	1
	No Mobile Site	Most other websites have mobile sites.	0
	Up Arrow to Go Home	Bad: Up arrows are rarely ever associated with return to home in other UIs.	3
Error Prevention	Automatic. Real-Time Spell Checker	Good: Spelling errors are easily noticed and can be avoided before compilation.	0
	Automatic, Real-Time Syntax Checker	Good: Some syntax errors (e.g. extra braces) can be avoided before causing a compilation error.	3
	Common Save Shortcut = Compile	Good: Ctrl + S (on Windows/Linux) causes the system to compile the code, rather than save the webpage, which is a common error when editing code in a web browser.	0
	Autosave	Good: Autosave is automatically enabled, preventing users from quitting editing without the changes being saved.	0
	Login Field Focus Feedback	Good: Login fields have obvious, but not distracting focus feedback (thin blue highlight), which helps users see where they are inputting text and prevent errors.	0
	Register Password Check	Bad: Register form has no way to check password (e.g. input match or reveal button), so input errors are unknown and difficult to reverse.	1

Table 3 (Continued)

Recognition Rather than Recall	Menu	Good: Offers many options that are recognisable and useful, but most users would unlikely be able to recall the name of what they are wanting.	0
	File-Tree Action Buttons	Good: Provide tooltips on hover.	0
	Compile Button Accelerator Tooltip	Good: On hover of the "Recompile" button, a tooltip suggests the keyboard shortcut to be used as an accelerator. This promotes recall for intermittent users who may forget the shortcut.	0
	Share History Suggestion	Good: Sharing to recent collaborators is suggested, reducing cognitive load for the task of sharing.	0
	Error Notifications	Good: Error messages cause notification number to appear next to link to find them.	0
	Front Page Get Started Buttons	Good: Multiple front page get started buttons mean user does not have to remember a single location after reading information.	0
	Projects Selection Page	Good: Lists all projects ordered by how recently modified they are, offering users to recognise the project they want rather than remember the name of the project they wanted.	0
	Templates Selection Page	Good: Lists available templates in sections, making it easy to recognise the type of template needed.	0
	Comment & Chat Preservation	Good: Collaborating users' comments and chat messages are preserved while system is not active. This helps users recognise other users' ideas and suggestions rather than remember them.	0
	Real-Time Syntax Suggestions	Good: LaTeX has many features, and syntax prompts help reduce cognitive load for the user, instead relying on recognition.	3
	Compile Settings Not Included In Menu	Bad: Compile settings are included in their own menu instead of the generic one, meaning users need to recall which menus are for which action, rather than them being recognised with a visual aid.	0
	Hidden Tools	Bad: Many tools are hidden or hard to find in the editor and are unusable unless users can recall how to access them. Examples include Find & Replace tool, PDF resizing, hiding the file-tree, edit title button, and compilation settings. This also crosses over with the "User Control and Freedom" heuristic.	2
Flexibility and Efficiency of Use	Compile Button Accelerator	Good: The frequently used "Recompile" button has an easy keyboard shortcut to be used as an accelerator, but the button is still easy to access for both experienced and beginner users. The accelerator is also suggested on button hover, which promotes recognition rather than recall for intermittent users.	1
	Full Screen Buttons	Good: Fullscreen buttons for the code editor and PDF viewer work allow for user to focus when needed.	0
	Visual Options in Menu	Good: Some options, such as font size and color scheme, offer users more ways to use the program (e.g. increased font size for a smaller screen).	0
	Templates	Good: Provided templates make it easy for new users to learn by example, and for experienced users to skip any tedious setup processes.	1
	Template Search	Good: Provides a quick way to find templates directly related to the users need.	0
	No Mobile Site	Bad: Loading the LaTeX editor on mobile loads a desktop site. This site is difficult to traverse and use, especially for a small mobile device.	0
	Menus Don't Collapse	Bad: Menus don't collapse, even when screen/section sizes hide some actions.	0

Table 3 (Continued)

Aesthetic and Minimalist Design	Chat and Comments	Good: Chat and comments are minimalistic and only show necessary information, providing a high signal-to-noise ratio.	0
	Logs & Output Files	Good: This section may not be useful to all users and so is kept out of the way, however it is implemented in a clean and concise manner for those who may use it.	0
	Editing Page	Good: Majority of screen taken by editor and viewer. Relatively focused design.	3
	Login Page	Good: Focused design with a high signal-to-noise ratio.	2
	Button Text Removal on Screen Size Reduction	Good: Buttons positioned at the top right of the screen lose their text, giving a larger signal-to-noise ratio, which is important for small screens. Unfortunately this decreases the match between system and real world.	0
	Dots on Dividers	Bad: Sets of four dots on section dividers are meaningless, which could be confusing to users. The dots could also be distracting as they are a relatively bright colour in comparison to their background.	0
	Create Account Button	Bad: The button used to register an account is very small on the login page.	0
	Front Page Top Banner	Bad: Front page top banner has a low signal-to-noise ratio, and is very busy. This could confuse users on where to go from here.	0
	Tooltips	Bad: Some tooltips cover other important text.	0
Help Users Recognize, Diagnose, and Recover from Errors	Error Messages	Good: Error messages are easy to locate, informative and are colour-coded to make things easy for the user to debug.	1
	Highlighted Error Sections	Good: Errors are highlighted in the code editor, and informative symbols with tooltips are available, helping users recover from errors more easily.	0
	Ability to Download Current Version	Good: Allows users to recover from large sets of errors relatively easily, provided preparations are made.	0
	Login Errors	Good: Login errors are logical and informative. They also provide a direct route to reset your password, facilitating UX.	0
	Register Password Error Feedback	Good: Error provided is logical and informative.	0
	Login Field Retention	Good: Login fields are remembered after a failed login. This is helpful for users to easily recover from an error.	0
	Errors	Bad: A few rare errors had no easy way out and caused me to refresh the page.	0
	Action History Cleared on Refresh	Bad: Mistakes made shortly before system close are difficult to undo, especially if they are on a large scale and no version has been downloaded.	0
Help and Documentation	Expansive Documentation	Good: Documentation for many different styles of users and use cases, including multiple languages.	2
	"Learn LaTeX in 30 minutes"	Good: Beginner users don't need to trawl through technical documentation to get started.	1
	Documentation Search	Good: Search is excellent and provides suggestions while typing, which provides recognition rather than recall and also an easy way to traverse the documentation. Many search results are questions, which are easy for users to see if their question will be answered on that page.	1
	Help & Documentation Content	Good: Content is understandable even for beginner users, and also provides step-by-step guides for issues to specific problems.	0