**Heuristic Evaluation**

**1 Strive for consistency.**  
Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

The navigation bar will be situated at the top of every webpage of the website which will contain the logo of the application on the left hand side(which when clicked on takes the user to the homepage), The **about** section to the right of the logo, an additional **options** dropdown to the right of the **about** button and if the user hasn’t logged in, then the login/signup buttons will be situated at the right hand side and if he has logged in then the user details will be displayed as a dropdown menu when he/she clicks the user icon.

**2 Enable frequent users to use shortcuts.**  
As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

If at any point the user wishes to go to the home screen, he/she need only click the logo situated on the navigation bar to go to that webpage instantly.

Conventional keyboard shortcuts(only applicable on desktops and laptops)(along with UI options) to undo, redo, save etc. can be used to increase the pace of interaction between the user and the application.

**3 Offer informative feedback.**  
For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

Always show/highlight where the AirPen’s pointer is on the board when the user is using their finger. Any finger movement by the user has to be clearly displayed on the file in front of the user.

Show a pop-up indicating to the user to save the file if the user tries to quit the application/open a new file without saving. The user may also choose to quit without saving.

Show a status window while saving the file and display a success message on another window upon successfully saving the file. If the file couldn’t be saved, then display a simple error message explaining why it failed.

**4 Design dialog to yield closure.**  
Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

Whenever the user logs in, there should be a message indicating it was successful.

Show a status window while saving the file and display a success message on another window upon successfully saving the file. If the file couldn’t be saved, then display a simple error message explaining why it failed. Similar messages on small windows can be displayed when trying to load a file.

**5 Offer simple error handling.**  
As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

Simple error handling procedures will make AirBoard very user-friendly. Procedures to handle invalid entries made while a user registers or tries to login can provide the necessary cues to correct their details that have been erroneously entered. Popups on the screen will help guide the user to fixes and provide more details on the matter. To make the application error-prone, features or options not accessible or applicable at a certain stage will not be shown to prevent the user from encountering unwanted errors. The application will not crash unless a fatal error has occurred.

At all given stages, instructions will be provided to users, along with a reason for the error to occur as well as clear-cut steps to mitigate them. This will allow greater control to users when it comes to handling the error and also provide them quick and easy fixes to the error at hand.

**6 Permit easy reversal of actions.**  
This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

AirBoard will not only feature a host of canvas and marker choices but also an eraser tool, which will allow users to quickly erase any erroneous strokes made. Additionally, users will also be able to adjust the size of this tool to help them erase larger regions faster or smaller regions with great precision. Users will also be able to choose a stroke on screen with their mouse and delete it.

AirBoard will also have the ever-useful undo as well as redo options, allowing users to quickly remove a stroke made and get back to drawing.

**7 Support internal locus of control.**  
Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

Common operations such as undo as well as delete will be keyboard mapped to convenient (but also customizable) shortcuts such as Ctrl+Z and Del. Users will also be given the choice to set keyboard shortcuts for quickly shuffling between choices of canvas, or picking a colour. Thus, a user will be able to customize AirBoard such that at most two devices (a target and either a keyboard or a mouse) can be used to operate the application.

AirBoard also keeps track of a user’s drawings made, and a user can easily get back to a drawing and update or delete it if necessary.

**8 Reduce short-term memory load.**  
The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

The layout of AirBoard will be simple as well as efficient. All options and access to features will be provided from a single screen. This will greatly reduce latency since there will not be any requirement of switching screens to access other features. Options to choose colours as well as canvas will be overlayed on the screen itself with visually suitable cues, therefore eliminating the need for a user to look for necessary features.

Panels will be made for features and they will be presented in an easy to access as well as in constant view of the user so they can quickly monitor other panels while drawing. The panels containing the features will be intuitive and all information will be relayed in an easily understandable manner.