* Implement GET call for *ID* from ***ENV*** platform
  + Use JavaScript dispatchEvent() and a JavaScript CustomEvent called ‘*customName*’ to interface with ***ENV***
  + Dynamically loads the context from ***ENV*** or defaults to standalone environment ID when calling the CustomEvent called ‘onSharedRequestCallback’
* Download *summary* button
  + Implemented API call with UI button on the (1) **admin** page and (2) **scoresheet** page
  + This features allows users to complete the MVP lifecycle and *generate summaries*, and additionally for all **scoresheets** (in the case of the **admin** page)
* Implement GET call for user information from ***ENV*** 
  + User Information is available from the ‘*customName*’ CustomEvent (referenced in the first bullet point)
  + This user information provides basic role based access when in ***ENV*** – the **admin** and **planning features** are only accessible for the *admin role*
* Implement Bug Fixes
  + Handled a production only UI scaling issue due to integration of *application with the* ***ENV***. This prevented data entry on pages.
  + Switching pages persists the selected **scoresheet** when navigating
  + Handle errors with better error messages – e.g. if the API is providing a good error message (like for uploading a *scoresheet*), then the UI propagates the error message to the user.
  + When editing input fields, pressing enter saves the input, and pressing the escape key cancels the save process
  + Changed the text of a warning message *to the specific use case*
  + Handled the *context for the* ***scoresheet*** in production properly (correctly managed the asynchronous call to the production API).
  + Sorted ***scoresheets names*** alphabetically
  + Changed the hover text to be a better UX message
* Implement Other Enhancements
  + Added a help page with user interface interaction descriptions, and a template **scoresheet** and instructions on how to fill it out properly.
  + Added a warning dialog when deleting a **scoresheet**
  + Added CSS styles to indicate alternating rows (banding) in presenting a set of **scores** under a single **section**
  + Added ability to modify the standard completion amount percentage for a **section**
  + Added GET request to load **scoresheets** based on the *context*
  + Improved the Landing Page (Home Page) with the *branding*