**Team Members:** Bryan, Cohen, Emily, and Jason

**Project title:** Pantry Pallet

**Project Description:** Application that allows users to find a recipe based selected ingredients and pairs the recipe with a wine.

**User Story:**

As a person who never has a stocked refrigerator, I want to enter random ingredients to receive a recipe based on those ingredients and a potential wine pairing, with option to see wine scores and price.

**APIs:**

<https://spoonacular.com/food-api/docs#Search-Recipes>

<https://spoonacular.com/food-api/docs#Get-Recipe-Information>

**Project Guidelines:**

1. Must use at least two server-side APIs
2. Must use a CSS framework OTHER than Bootstrap
3. Use at least a new third-party API
4. Must have a polished, mobile-first UI
5. Must meet quality coding standards (indentation, scoping, naming, etc.)
6. Does NOT use alerts, confirms, or prompts (look into modals)
7. Must be deployed to GitHub Pages
8. Must be interactive (accept and respond to user input)

# Document version

|  |  |  |
| --- | --- | --- |
| **Version** | **Changed By** | **Description** |
| 1.0 | Jason Miller | Initial draft |
| 1.1 | Jason Miller | Updated with group feedback and wireframe added |
| 1.2 | Jason Miller | Requirements/overview updates |

# Suggested Approach

**Spoonacular** allows a user to input an ingredients list and will return both recipes and wine recommendations.

The user inputs up to five ingredients.

The five ingredients are used in the spoonacular call.

Spoonacular then returns a list of objects which are together individually as:

* **Recipe title**
* **Recipe link**
* **Wine pairing**

To the user, a recipe renders as a title and image, along with a wine pairing, rating, and score (if a pairing is available) and a link to the original source page of the recipe.

When the user clicks a recipe link, she is taken to the recipe website.

Based on that approach, we would need the following core elements:

|  |  |  |
| --- | --- | --- |
| **Element/Function** | **Name** | **Description** |
| Array | ingredList | Holds ingredients input by user |
| Input | ingredInput | Allows user to input one ingredient at a time (up to five) |
| Para | ingredDisplay | Displays the list of added ingredients to user |
| Button | ingredAddBut | Allows user to add ingred entered into ingredInput and show in ingredDisplay |
| Button | ingredClearBut | Clears all ingreds from inputDisplay |
| Button | ingredSubmit | Launches both APIs and clears the ingredDisplay |
| Array | foodWineList | Holds objects from spoonacular return |
| Div | ID = recipeReturn | Empty div that is auto-populated upon API recipe return |
| Listener | For ingredAdd | Upon click for add button |
| Listener | For ingredClear | Upon click for clear button |
| Listener | For ingredSubmit | Upon click for submit button |
| Function | ingredAdd | Allows user to add ingred entered into ingredInput and show in ingredDisplay |
| Function | ingredClear | Clears all ingreds from inputDisplay |
| Function | ingredSubmit | Launches both spoonacular APIs and displays results |

Based on this description and these fields, we would need to assign the following tasks:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Type** | **Assign** | **Notes** |
| Finalized requires | All |  |  |
| Create mobile layout wireframe | Front end |  | Links not images? |
| Create responsive layout wireframe | Front end |  | Images not links? |
| Select color palette | Front end |  |  |
| Select CSS framework | Front end |  |  |
| Build additional/custom CSS entries | Front end |  |  |
| Build html interface | Front end |  |  |
| Create API call to spoonacular | Back end |  | Includes local storage of returns; returns 5 recipes |
| Create API call to wine-searcher | Back end |  |  |
| Create function - renderPage | Back end |  |  |
| Create function/listener - ingredAdd | Back end |  |  |
| Create function/listener - ingredClear | Back end |  |  |
| Create function/listener - ingredSubmit | Back end |  | Includes two API calls to spoonacular |
| Unit testing – front end | Front end |  |  |
| Unit testing – back end | Back end |  |  |
| Device testing - mobile | TBD |  |  |
| Device testing – laptop | TBD |  |  |
| Device testing - desktop | TBD |  |  |
| Create presentation deck | TBD |  |  |
| Practice presentation | All |  |  |

# appendix a: API DOCUMENTATION

<https://spoonacular.com/food-api/docs#Search-Recipes-by-Ingredients>

<https://spoonacular.com/food-api/docs#Get-Wine-Pairing>

<https://api.spoonacular.com/recipes/716429/information?includeNutrition=false>

Account Key ([jaericm@yahoo.com](mailto:jaericm@yahoo.com) / bootcamp): acc2d918d19f494f9490b92a1b73fc4d

**Sample recipe call and return:**

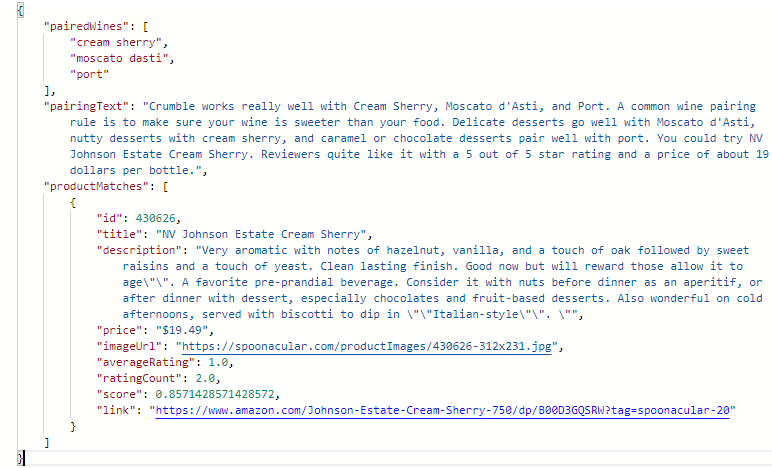
<https://api.spoonacular.com/recipes/findByIngredients?ingredients=apples,+flour,+sugar&number=1&apiKey=acc2d918d19f494f9490b92a1b73fc4d>



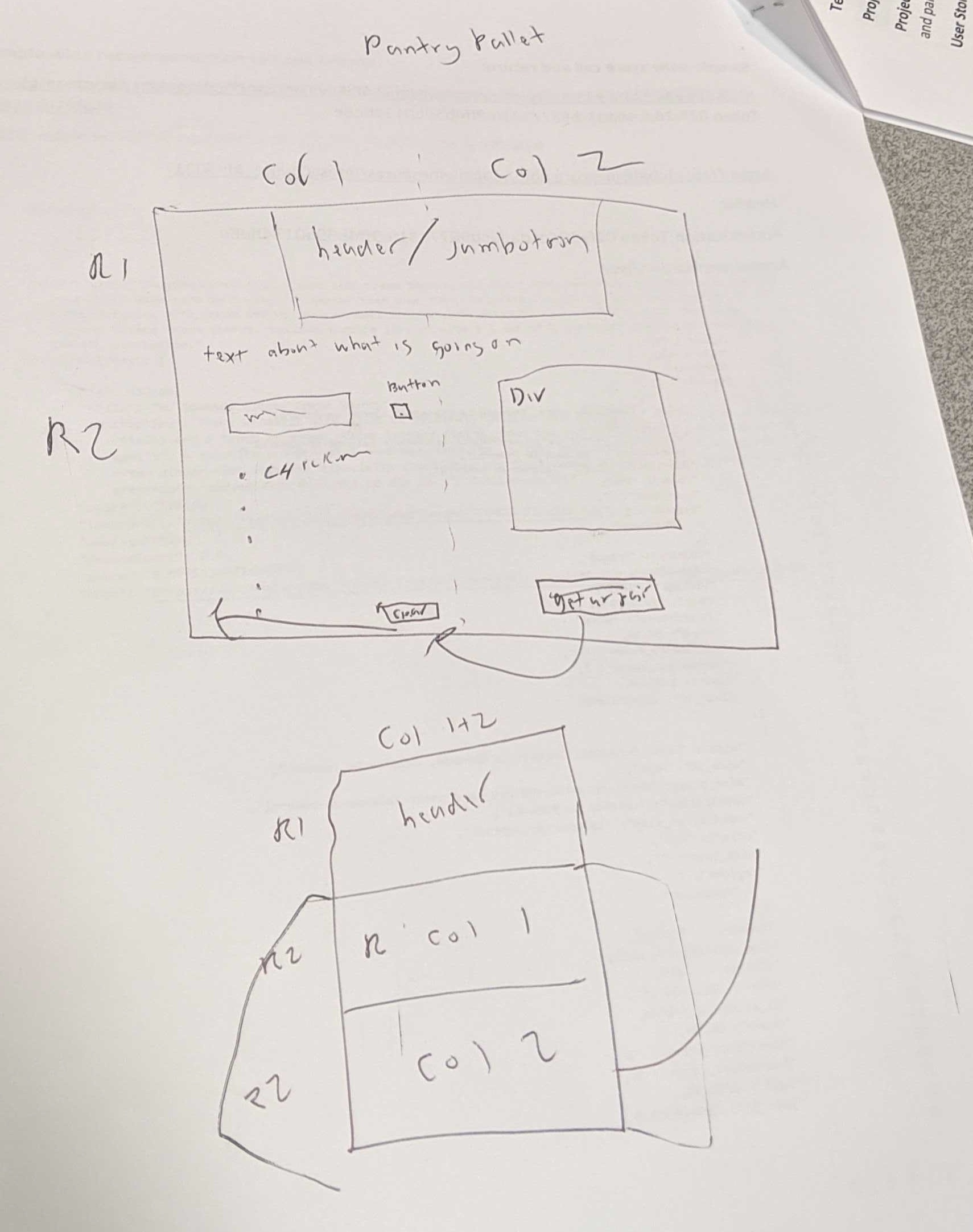
**Sample wine recommendation call and return:**

<https://api.spoonacular.com/food/wine/pairing?food=Crumble&apiKey=acc2d918d19f494f9490b92a1b73fc4d>

NOTE: maximum price could also be entered as an optional variable



# appendix B: WireFrames and color palette



Red rose Wine color palette created by taylorrmcv that consists

#ffb9b9

,#ee7272

,#a31818

,#6d0202

,#360000