Integration- Bottom-Up & Sandwich

For our project LifeTracker, we used two different Integration Strategies. We used Sandwich Integration and Bottom-Up Integration. Bottom-Up Integration is used for individual work on components and Sandwich Integration is used for combining individuals' works. Functions for Budget component, grocery component, and weight component were coded first and then we put them together to construct components. CSS files were edited after the components were done. To combine every individuals' works together, we implemented and integrated logic artifacts top-down and operational artifacts bottom-up. To put the components together, first we created a home component and put necessary information and then we added other components on the home page.

The first reason why we used Sandwich Integration to combine the components is because it made the most sense to use Sandwich Integration. It has advantages of fault isolation, catching major design faults early, and testing the potentially reusable code artifacts adequately. It worked great for combining works because each member had their own components which were made bottom-up and the home component was built top-down. So essentially everything is working towards each other. The reason why we used Bottom-Up Integration is because it seemed like it was the simplest choice for each component. Bottom-Up Integration also has advantages on fault isolation and testing the potentially reusable code artifacts adequately. It worked great for individuals' works since we start with functions and use them in the components. Individually, we built components bottom-up and we put them together. After all the components were finished, we built the home component top-down and linked the budget, grocery, and weight components.