**Sprint 1 Artifacts**

**Requirements Artifact For Task Display**

**System Overview:**

The website allows users to switch between two displayable pages. The initial page will contain current tasks and the fields to create new ones/update old ones, and the second page will contain completed tasks and fields to update them.

**Functional Requirements:**

User Story:

As a user, I can switch between a current task page and a completed tasks page by clicking on one of two tabs so that I can track what tasks I have to complete and what is already finished. These tabs will display those tasks corresponding to their respective statuses.

* 1 - Tab Switching Buttons - Skylar Franz
  + 1.1: The application will initially display the current tasks page.
  + 1.2: The application will have a button that navigates to the current tasks page.
  + 1.3: The application will have a button that navigates to the completed tasks page.
* 2 - Task Displays - Skylar Franz
  + 2.1: The application will display all of the user’s current uncompleted tasks stored in the database when it loads the current tasks page.
  + 2.2: The application will display all of the user’s completed tasks stored in the database when it loads the completed tasks page.
* 3 - Task Creation Field - Skylar Franz
  + 3.1 - The application will display a text input field to ascribe a task’s name in.
  + 3.2 - The application will display a text input field to ascribe a task’s description in.
  + 3.3 - The application will display a button for submitting the task for creation.

All functional requirements were met.

**Requirements Artifact for Task State Management**

**System Overview:**

The website is connected to a backend database, set up using Laravel, that tracks current and completed tasks (names and descriptions). Through the website, the user can create tasks, delete tasks, mark tasks as completed, or mark completed tasks as incomplete.

**Functional Requirements:**

User Story 1:

As a user, I can create tasks and delete tasks in order to track things that I need to do.

* 1 - Task Creation - Skylar Franz
  + 1.1: Clicking the submit button will create an entry in the database containing the task’s information as taken from the input fields.
  + 1.2: Clicking the submit button without giving the task a name should result in the database not creating a task.
  + 1.3: Clicking the submit button without a task description, timer, or schedule will still result in creating a task.
  + 1.4: Task Creation functionality should be available in only the current tasks page.
* 2 - Task Deletion - Skylar Franz
  + 2.1: Clicking the delete button on an active task will delete its entry from the database.
  + 2.2: Task Deletion functionality should be available in both the current tasks and completed tasks pages.

User Story 2:

As a user, I can mark a current task as complete when I have finished it to keep track of my successfully finished work.

* 1 - Task Completion - Skylar Franz
  + 1.1: Clicking the complete button on a task should switch its status in the database to be “complete.”
  + 1.2: Task completion functionality should be available only in the current tasks page.

User Story 3:

As a user, I can mark a completed task as incomplete so that I can undo a mistaken input that marks a task as complete.

* 1 - Task Incompletion - Skylar Franz
  + 1.1: Clicking the incomplete button on a task should switch its status in the database to be “incomplete.”
  + 1.2: Marking a task as incomplete should be functionality that is available only in the completed tasks page.

All functional requirements were met.