

Class 8: Sprint Planning – GitHub Issues

1. Introduction to Homework 3
2. UI Component API Design
3. GitHub Issues
4. Studio

Copyright 2026, Kyle J. Harms

Activity: Create Homework 3 Repo

We need to grant you additional access in your Homework 3 repository.

Create it now from the **Course Resources** page in Canvas.

Review: v-model

Bind user input data from input elements (i.e. `<input>`, `<textarea>`, `<select>`) to a reactive variable using the `v-model` directive:

```
<script setup>
const checked = defineModel(false);
</script>

<template>
  <div>
    <input type="checkbox" v-model="checked" />
    <p>Checked: {{ checked }}</p>
  </div>
</template>
```

Review: TailwindCSS

TailwindCSS is a utility-first CSS framework for rapidly building custom user interfaces.

Example:

```
<button class="bg-blue-500 text-white font-bold py-2 px-4 rounded">  
  Next  
</button>
```

Homework 3

Homework 3: Prototype UI Component Library

Using only **Vue.js**, **TailwindCSS**, and *optionally* **daisyUI**, implement a user interface component library.

User Interface Component Library: A collection of reusable UI components that can be used to build user interfaces.

Example: [MUI](#), [Hero UI v2](#), etc.

In **App** include a demo gallery of each UI component.

Homework 3: UI Components

The user interface components you create should be **reusable** and **configurable**. Each component should have a **clear purpose** and **API**.

```
import {Button} from "@heroui/react";

export default function App() {
  return <Button
    color="primary"
    variant="bordered"
    onPress={() => console.log('Button pressed!')}
  >Button</Button>;
}
```

Activity: Component API Design

Working with a **peer**, study the components for Hero UI v2:
<https://v2.heroui.com/components>.

On your **handout**, design the API for a **button**, **switch**, and **card** components.

Note: In your prototype, you will implement **only** the **necessary** functionality for each component, and no more. (No “extra” features.)

Gotcha: Slots

Slots are a way to pass content from a parent component to a child component. They allow you to create more flexible and reusable components.

App.vue

```
<script setup>
import Card from 'Card.vue'
</script>
<template>
  <Card>
    <template #header>
      <h2>Card Header</h2>
    </template>
    <template #body>
      <p>Card Body</p>
    </template>
  </Card>
</template>
```

Card.vue

```
<script setup>
</script>

<template>
  <div class="border border-gray-300 rounded-lg">
    <header class="bg-gray-100 p-4">
      <slot name="header"></slot>
    </header>
    <main class="p-4">
      <slot name="body"></slot>
    </main>
  </div>
</template>
```

Gotcha: Component Naming

When creating components, use **PascalCase** for the component name. **lowercase** component naming is reserved for **HTML elements**.

HTML Element:

```
<button>Click Me</button>
```

Vue.js Component:

```
<script setup>
import Button from 'Button.vue'
</script>

<template>
  <Button>Click Me</Button>
</template>
```

Sprint Planning

GitHub Issues

Review: Sprints

Scrum sprints are short, time-boxed periods (often a week or two, but less than a month) where teams focus on completing a set amount of work, enabling incremental delivery.

Source: [Atlassian](#)

Planning Sprints

A set of **tasks** need to be identified, prioritized, and assigned to a **sprint**.

Often in software development, these tasks are represented as **GitHub Issues**.

Activity: Create GitHub Issues

Create an issue to create a component for the **button**, **switch**, and **card** components in your Homework 3 repository.

These issues will be part of your first sprint for Homework 3.

Activity: Button Issue

1. Create a button component in your Homework 3 repository.

Note: DaisyUI is available for use in your repository.

2. Commit your changes using a commit message that references the issue number (e.g. `#1`).

3. When you are finished with your changes, close the issue.

If you want to close the issue automatically when you push your changes, use the following commit message:

```
implement button component (closes #1)
```

Studio: Homework 3

For each component:

1. Create an **issue** for the component,
2. Implement the component in your Homework 3 repository.
3. Commit **only** the code for that issue.
4. Close the issue.

What's Next

Homework 3, Sprint 1 - Due Monday, September 21, 2026 at 11:59pm

Following Week: Oral Exam I