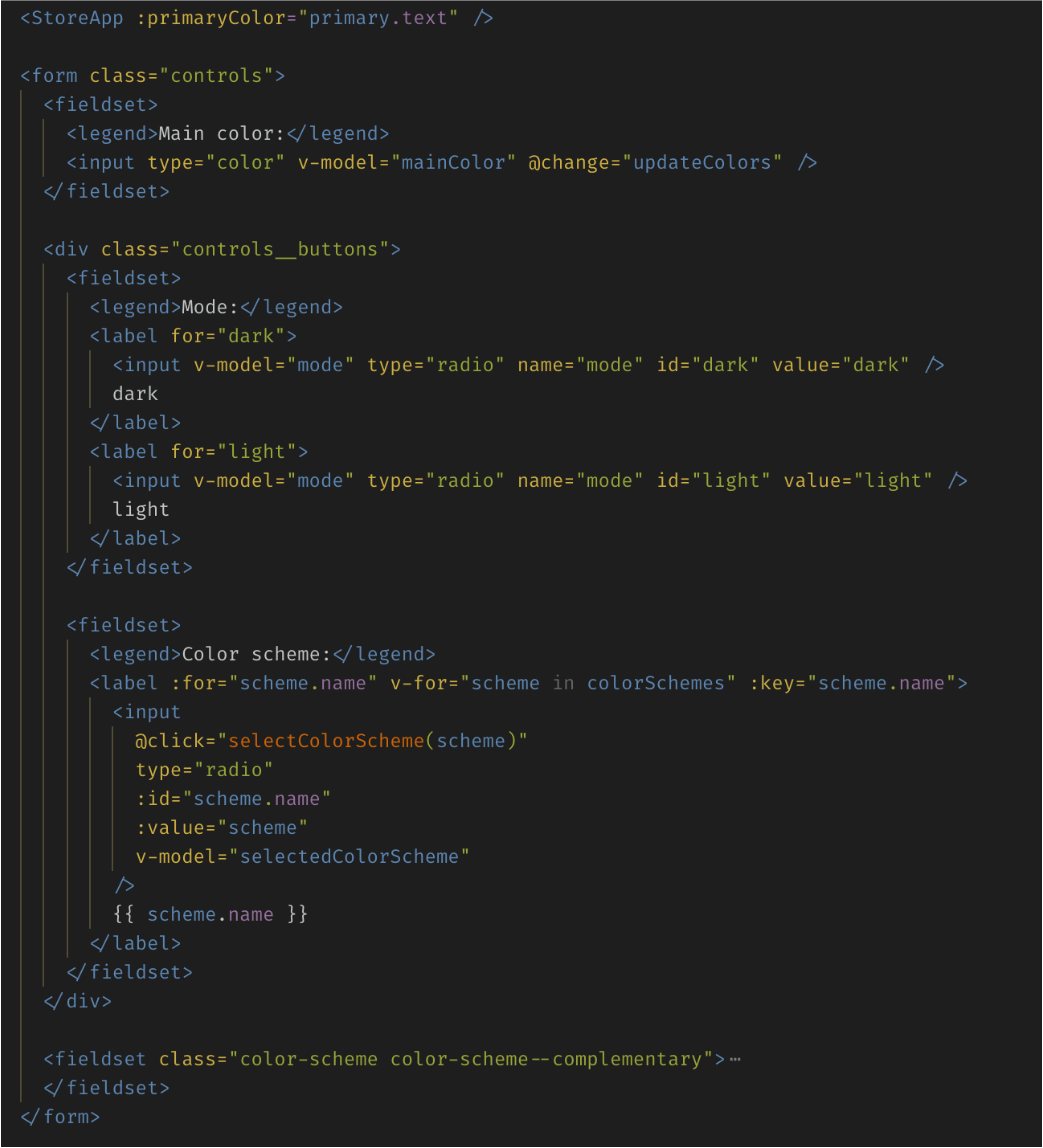
# GENERATING ACCESSIBLE COLOR SCHEMES



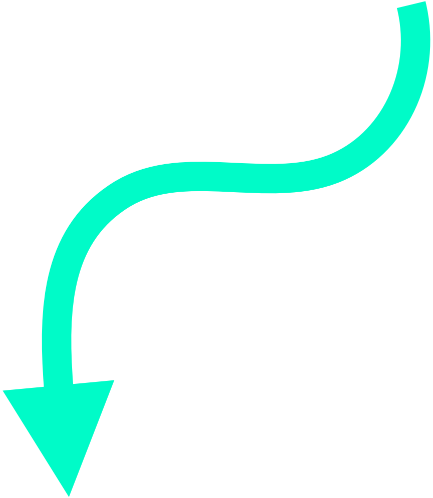
```
<StoreApp :primaryColor="primary.text" />

<form class="controls">
  <fieldset>
    <legend>Main color:</legend>
    <input type="color" v-model="mainColor" @change="updateColors" />
  </fieldset>

  <div class="controls__buttons">
    <fieldset>
      <legend>Mode:</legend>
      <label for="dark">
        <input v-model="mode" type="radio" name="mode" id="dark" value="dark" />
        dark
      </label>
      <label for="light">
        <input v-model="mode" type="radio" name="mode" id="light" value="light" />
        light
      </label>
    </fieldset>

    <fieldset>
      <legend>Color scheme:</legend>
      <label :for="scheme.name" v-for="scheme in colorSchemes" :key="scheme.name">
        <input
          @click="selectColorScheme(scheme)"
          type="radio"
          :id="scheme.name"
          :value="scheme"
          v-model="selectedColorScheme"
        />
        {{ scheme.name }}
      </label>
    </fieldset>
  </div>

  <fieldset class="color-scheme color-scheme--complementary">⋯
  </fieldset>
</form>
```



```
updateColors() {
  this.primary = this.generateTextAndBg(this.mainColor);

  this.updateColorVariable("--primary-text-color", this.primary.text);
  this.updateColorVariable("--primary-bg-color", this.primary.bg);

  // ...
}
```

```
generateTextAndBg(color) {
  let bg = Color(color);
  const text = bg.isLight() ? Color("#000") : Color("#fff");
  let constrast = text.contrast(bg);

  while (constrast < MIN_CONTRAST) {
    bg = text.isLight() ? bg.darken(0.5) : bg.lighten(0.5);
    constrast = text.contrast(bg);
  }
  return { text: text.string(), bg: bg.string() };
}
```

[redacted]

```
updateColors() {
  this.primary = this.generateTextAndBg(this.mainColor);

  this.updateColorVariable("--primary-text-color", this.primary.text);
  this.updateColorVariable("--primary-bg-color", this.primary.bg);

  // ...
}
```

```
updateColorVariable(colorName, newValue) {
  document.documentElement.style.setProperty(colorName, newValue);
},
```